MINIMUM TIME CONTROL WITH TRAJECTORY REASSIGNMENT FOR MULTI DEGREE-OF-FREEDOM SYSTEMS

By

WILLIAM PRATT MOUNFIELD, JR.

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

1985

In memory of William Pratt Mounfield

ACKNOWLEDGEMENTS

I would like to express my appreciation to the Department of Mechanical Engineering, University of Florida, for funding my research. I also thank the General Electric Corporation, Drive Systems Division, for granting me an academic leave of absence to continue my graduate research. I extend my appreciation to Dr. Delbert Tesar for encouraging me to pursue a doctoral degree. In addition, I thank Dr. Alexander Meystel for posing the trajectory reassignment problem.

I would like to thank the following members of my committee, Dr. Robert E. Gaither, Dr. Joseph Duffy, Dr. Gary K. Matthew, and Dr. Kermit N. Sigmon. Finally, I would like to thank Dr. George N. Sandor for his encouragement and confidence without which I would not have completed this degree.

TABLE OF CONTENTS

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

MINIMUM TIME CONTROL WITH TRAJECTORY REASSIGNMENT
FOR MULTI DEGREE-OF-FREEDOM SYSTEMS

By

WILLIAM PRATT MOUNFIELD, JR.

August 1985

Chairman: George N. Sandor
Cochairman: Joseph F. Duffy
Major Department: Mechanical Engineering

This dissertation addresses the problem of obtaining minimum time control of a multi degree-of-freedom system with midstress trajectory reassignment. A general model for N-degree-of-freedom systems is developed that includes the actuator dynamics. An existing feedback linearizing and decoupling transformation (FLDT) is used to force each degree-of-freedom to be a linear triple integrator system. The FLDT is used here to obtain simple solutions to the time optimal control problem. An example is developed for a two-degree-of-freedom system demonstrating effective linearization and decoupling.

The existing time optimal positioning control (TOPC) for static endpoints is used here as a means for analyzing an approach in construction and concatenation of reassignment trajectories. The problem of reassignment is then

analyzed by the segmentation of all nominal tra[illegible] into general cases of active state variable constraints. This permits analysis of when a reassignment can occur within a nominal trajectory, involving the division of each nominal trajectory into seven velocity sectors. When reassignment occurs in different sectors, different trajectory solutions are required. The solutions to reassignment could be one of continued forward motion, stopping short, or the reversal of the forward motion. The reassignment trajectory then assumes the role of a nominal trajectory in which further reassignment may occur. A one-degree-of-freedom example is given in which reassigned motion requires the reversal of the nominal trajectory.

The decision of which reassignment input jerks to apply to the triple integrator system is shown to be effected by incremental additions in areas under simple velocity profiles. These velocity profiles, produced from the known actuator performance parameters and manipulator dynamics, are similar to those used in the TOPC.

The structures of the various solutions are intended to accommodate real time implementation in an controller faced with alternative decisions for minimum time trajectory planning. The construction of jerk inputs for reassignment trajectories is developed to provide a universal method based upon dynamic parameters of maximum acceleration and actuator motor constants.

CHAPTER 1
INTRODUCTION

This dissertation presents a technique for obtaining a solution to minimum time control with trajectory reassignment of multi degree-of-freedom systems. Multi degree-of-freedom systems are those characterized by nonlinear coupled differential equations and whose position coordinates depend upon independent variables. Examples of these systems are multilink manipulators and autonomous mobile systems. Each of these systems has the opportunity for an assigned positional task to require the minimum time motion of one or more of its degrees of freedom. Further, during minimum time motion in the degrees of freedom it is realistic to require a new final position of the system different from that task which was originally assigned. This dissertation describes a technique for the real time solution of the reassignment problem.

Problem Statement

The problem of trajectory reassignment during minimum time motion of multi degree-of-freedom systems is a portion of the control system outlined by A. Meystel in "Intelligent Control of a Multiactuator System", [15,Meystel,1982]. Intelligent control being that control which

would replace the human mind in making decisions, planning control strategies, and learning new functions whenever the environment does not allow and does not justify the presence of a human operator. [23,Sardis,1977,p4]

The emphasis of controllers to be used in these systems is those that provide real time control during which alternative solutions normally are required. The function of optimal control should be performed at the lowest level of an intelligent controller with the ability of the optimal controller to respond to any task in real time. The specific solution offered in this work is to define a technique for the minimum time motion for systems when the positional task are reassigned.

The multi degree-of-freedom systems described in this dissertation are those of multilink manipulators, however the work could be extended for other multiactuator systems. The models are developed for serial manipulators of known geometric configuration and known dynamic constraints. The dynamic model for the manipulators include that of the actuators since the actuator's limitations contribute significantly to the optimal motion of the manipulator. The inclusion of the actuators dynamics and its constraints while increasing the knowledge of the actual manipulator also increases the complexity of the required optimum inputs for minimum time motion.

Allon Guez in "Optimal Control of Manipulators" [10,Guez,1983] developed a control technique for the minimum

tire motion of manipulators for static endpoint moves. Guez was one of the first to incorporate actuator dynamics into the control strategy. He realized, that due to the complexity of the manipulator and actuator dynamics, some transformation of the state variables was necessary in order to find a closed form solution to the minimum time problem. Therefore the so called "feedback linearization and decoupling transformation" (FLDT) [10,Guez,1983] was used to force each of the links and associated actuators in a serial manipulator to be a time invariant linear triple integrator system. This linear system for each joint allowed Guez to use a closed form solution for minimum time motion with static endpoints from the previous work of [25,Smith,1966]. It should be noted that the difficulty in the solution is that of including the inequality constraints of the state variables resulting from actuator limitations. Guez called his method "Time Optimal Positioning Control" (TOPC).

The problem addressed in this dissertation is that in which the target endpoint of a minimum time move is reassigned in midstream during that move. If it is required that the new endpoint be reached also in minimum time, no closed form solution has been found to determine the new optimal inputs for the new move that also observes the state variable inequality constraints (SVIC). Therefore, what follows is a technique to allow the generation of such inputs based upon rules of allowed velocity, acceleration,

and input jerk and the method to merge or concatenate the new inputs with those inputs that are currently being applied to the manipulator. These new control inputs are those that can be labeled TOPC with Mid Stream Trajectory Reassignment (MSTR).

Literature Review

The literature reviewed in this section demonstrates that the subject of trajectory with minimum time motion for manipulators is an object for concern for many investigators. The general subjects to be discussed are whether or not minimum time planning is possible for a particular trajectory technique, whether or not the knowledge of the actuator dynamics (or even manipulator dynamics) is used, and what is the solution technique to be used to solve for the proper trajectory when it is a claimed "minimum time path". The literature includes supporting evidence as to the uniqueness of minimum time control with mid-stream trajectory reassignment. The review begins with trajectory planning solutions that are at least applicable to a trajectory planning problem and progresses to those that are the "state-of-the-art" for trajectory planning.

The majority of the trajectory planning techniques published to date are ones that do not account for the manipulator and/or actuator dynamics. Some recent examples of this open loop planning do include the use of computer aided techniques, that is to say, the application of

graphical display of wire frame or solid models of manipulators during a programmed move. Derby [6,1984], Dillmann [7,1983], Jacobs [12,1983], and Sjolund and Donath [24,1983] demonstrate examples of this technique which typically are geometry oriented and observe only kinematic constraints on the manipulators. These animation systems are not for the search or generation of minimum time paths, rather they are consistent with the testing of candidate robot programs that generate a path in the work space.

Taylor [27,1979] and Milenkovic [19,1979] offer two classic papers on trajectory planning which are the basis for the "most common form of manipulator control" [5,Castin and Paul,1984,p68] utilizing polynomials to provide interpolating points at some specified servo rates between specified positions. That is, "knot points" are chosen along a particular straight line path between two points in Cartesian space and a uniform velocity is chosen between the knot points with constant acceleration between segments [27,Taylor,p430]. The velocity profile is iteratively chosen (by adding more knot points) to satisfy the error criterion for maximum trajectory deviation and smooth acceleration between constant joint velocities for each knot point [4,Brady et al.,1982,p31]. The primary problem with this technique is the "tendency of such trajectories to undershoot or overshoot the desired position . . . and (the robot) is difficult to control with polynomial trajectories"

[5,Castin and Paul,p68]. Obviously no attempt is made for minimal time motion with regard to dynamic considerations within these techniques.

Two papers that offer interesting variations of the non-dynamic trajectory planning problem are one on APT (Robot Automatic Programming Tool) [1,Abbler et al.,1982], and a paper by Totani and Miyakawa [28,1981]. These two papers project the interaction of the robot with boundaries generated by surfaces of workspace objects. Even the manipulator is modeled by a collection of objects with certain spatial relationships and an attempt is made to move with smooth trajectories along boundaries which the manipulator does not penetrate. Neither of these techniques address minimal time issues or mid-stream reassignment, however, they may provide insight as to how to plan minimum time motion among moving obstacles.

The next level of knowledge for trajectory planners is the use of the manipulator's dynamics in trajectory simulation. The motion control of manipulators with dynamic considerations of the arm movements has been recognized as necessary in order to minimize the path errors generated by techniques using kinematic descriptions only. Ahmad and Besand [2,1984] and Vukobratovic and Stokic [30,1982], Vukobratovic et al. [31,1983] state that the kinematic and dynamics must be accounted for, and each propose a solution of the ensuing control problem design that uses this

knowledge. Vukobratovic and Stokic [30,1982] offer a design method for choosing a manipulator for a specified task based upon the dynamics and kinematics of a particular robot. The actual controller design is formalized [31,Vukobratovic et al.] where a linear quadratic regulator is designed in order to minimize a quadratic performance index of trajectory and control effort weights. Nominal desired trajectories for a manipulator are then synthesized based upon an approximate decoupled. The task of the controller is to follow this trajectory while minimizing the quadratic performance index. A significant part in the design of the controller is to attempt to decouple the nonlinear manipulator system by forming a set of independent subsystems which use a generalized external force vector acting on each of the subsystems. The local control law uses a linear feedback law to minimize this interaction, which Vukobratovic and his collegues admit [31,1983,p249] will be sub-optimal for large nonlinear interactions resulting from fast moves.

Ahmad and Besant [2,1984] also aim to minimize path errors due to trajectory design without dynamic considerations [p305]. They state, however, that the design of such trajectories which will "execute without producing actuator demands which are in excess can only be solved by simulation," [p306] due to the inability to separate the velocity and acceleration in the dynamic trajectory space equation.

Therefore they propose an adaptive control scheme to adapt the trajectory based upon the computation of the maximum trajectory velocity obtainable [p306]. Due to computational burdens, the implementation of this technique has been limited to ignoring the inertial coupling forces in the model. Yet Ahmed and Hassan did use knowledge of the manipulator dynamics and one limitation of maximum available joint torque to minimize path following errors.

The next category of papers also considers some of the actuator limitations in order to plan a trajectory. Anderson and Paul [3,1979] have an objective to drive the "slowest joint at its maximum speed [to] maintain coordination and minimize time" [p449]. However, Anderson and Paul use a linear model of the dynamics with coupling for computational considerations which would be applicable for lower joint velocity. The velocity constraints are "conservatively set" [p449] to minimize interaction, yet show no method for picking such.

One of the earlier papers for minimal time control by Kahn and Roth [13,1971] did address the time optimal control of a manipulator. Their solution to the minimal time problem was based upon the maximum principal of Pontryagin [p166] with bounded torque input to the system. The minimization of the resulting Hamiltonians was a solution to a set of twelve first-order nonlinear differential equations. Since a numerical approach is the only solution to

this two point boundary problem, with the initial and final states, Kahn and Roth proposed a suboptimal control which used a linear set of equations of motion by assuming the velocity is low so the coupling is low. The solution of the set of linear equations with a torque constraint was obtained from a variation of earlier work by Flugge-Lotz and Titus [9,1962,p152].

Steinmann and Klafter [26,1963] extended the limits on possible trajectory inputs by realizing that "large jerks in the joint variables are to be avoided" [p20.72]. They developed a sequence of five connected polynomials each of degree three or less to generate a straight line path for a 3-DOF manipulator. This scheme used the idea that in order to limit the jerk on the joints, a finite third derivative was necessary in the manipulator. However, the method of path generation does not use other possible limits that might exist, such as those on velocity or acceleration.

The concept, possibly originated by Kahn and Roth, of treating each DOF of the manipulator as a independent double integrator system was extended by Patzelt [21,1982]. The method of "dual mode" [p391] control for moves of great distances or approach without collision and moves of small distances with accurate following was observed as two different problems requiring two different control schemes. Both methods of control were based upon decoupling the links by means of feedback of an inverse system. Patzelt did not

extend his work to that of including the actuators where not only acceleration constraints, which he did include, but velocity constraints may exist. In order to help reject unmodeled disturbances and correct for inaccurate initial conditions during a minimal time move, the input to each joint was a function of position and velocity errors between a reference double intergrator system and the real manipulator.

The recent paper by Isaa et al. [11,1984] was the most accurate modeling paper found for this research. This effort placed its emphasis on the animation of manipulators that are based upon a nonlinear model of the links, connecting mechanism, actuators and their servo drives. The models even included discretization effects of using finite word length digital controllers. Since the primary task of this project was to develop the simulation of paths of robots, it was left up to the programmer to redefine a trajectory if limits of the actuators were exceeded. There was then no knowledge of the limits of the system used to generate the proper minimum time moves, when desired, and all work was to be performed off-line in order to check proposed paths. This simulation system would form an excellent system to check the validity of a minimum time move which should not require more of the actuators than they could deliver.

Kuno et al [14,1979] and Parzley [22,1978] address the simulation, generation and storage for future use of proposed minimum time paths. However, neither of these schemes address the true dynamic systems of the manipulator that could prevent such a trajectory from being realized. The storage of joint positions to duplicate that motion which has been generated with minimum time motion in mind is worth noting, although a formidable task due to the storage requirements.

The final group of papers reviewed represents the highest integration of manipulator dynamics and minimal time trajectory planning. Each should be studied in conjunction with the recent work of Allon Guez [10,1983], for each paper treats the problem of generating an initial minimum time trajectory off-line. A paper that was included in [10,Guez] by Lin, Cheng and Luh [17,1983] used a technique not unlike the kinematic method for generating a straight line trajectory by fitting cubic spline functions to knot points. Their's is an iterative search method that recognizes joint saturations but does not include a method of generating inputs with the constraints as a guide. Also, the motion to program is assumed to be along a preplanned straight line path. This method could be used to check a possible minimal time trajectory candidate produced by some other method.

A computer graphic simulator that does not include manipulator dynamic constraints is an iterative technique

proposed by Lau and Mahajan [16,1984]. This method scales down the velocities and accelerations of fast moving joints [p4.80] so as to minimize the required joint torque. The authors do recognize the fact that a candidate move could be divided into regions of long or short moves which meet increasing demanding constraints. These iterations are performed off-line on a preplanned set of points.

Toyama and Takano [29,1983], Dubowsky and Shiller [8,1984], and Castin and Paul [5,1984] each offer solutions to the specific minimal time trajectory generation problem, each with a different iterative technique. Toyama and Takano use a true nonlinear programming technique with a modified gradient projection extension. The model of each DOF is only second order but this method does address the problem of minimal time motion around a known static object.

Castin and Paul [5,1984] seek to minimize the shocks applied to a manipulator limiting the rise time of a square wave acceleration profile. The trapezoidal acceleration input has its slope assigned arbitrarily [p69], not based upon the actuator constraints. Each path segment is divided into two equal paths and the maximum attainable acceleration is not used symmetrically for both path segments. Clearly this is a suboptimal solution and there is no concept of a possible required constant velocity portion of the trajectory. The off-line iterative program to develop the

trajectories assumes that the manipulator is at rest for the final and initial points, so no reassignment is possible.

The last paper reviewed is that of Dubowsky and Shiller [8,1984] which, as others have assumed, places no jerk limits on the actuators. Even though claimed to be significantly faster than other methods, this technique did require one minute of cpu time on a PDP 11/44 for each one second move with known initial and final static states. This is not an on-line method that would allow for trajectory reassignment.

In all the papers in this review, the concept of possibly reassigning a minimum time trajectory during the nominal motion does not exist. No method presented or found to date other than that of A. Guez is quantitatively fast enough to allow a possible real time calculation of minimal time motion for an N-degree-of-freedom manipulator. The work presented in this dissertation will use the Guez model as a basis so that mid-stream reassignment of a minimum time trajectory is possible in real time.

CHAPTER II
MODEL DEFINITION AND LINEARIZATION

Model of Multi-Degree-of-Freedom System

In order for the realization of minimum time trajectory control with mid-stream reassignment to be possible as represented by this work, many assumptions must be met by the manipulator system. This chapter furnishes the background work that is assumed to be operational at least in the specification of performance if not the exact method of implementation. The minimum time controller must output its commands to sub-systems that appear to be linear, third order systems with globally decoupled dynamics from all other links. Further, there are assumed to be a priori known maximum performance constraints on the system's velocity, acceleration and allowable jerk input. Therefore, what follows in this chapter is a description of a method following that of and credited to "Optimal Control of Robotic Manipulators", by Allon Guez [10,1983], which forces the manipulator system to meet the necessary dynamic model specifications

The major assumptions for the manipulator are that for the following deterministic system

- each link is driven by a linear third order time invariant permanent magnet Direct Current motor

- the manipulator is an open loop serial chain of N rigid links with rotary or prismatic joints
- all kinematic and dynamic specifications of the system are known
- all external loads are known differentiable functions of the joint states and time.

Since the manipulator is a rigid serial open chain, a model may be written as follows

$$\underline{T} = J_1(\underline{\theta},t)\underline{\alpha} + \underline{l}(\underline{\theta},\underline{\omega},t) + \begin{vmatrix} \underline{\omega}^T G^1(\underline{\theta})\underline{\omega} \\ \underline{\omega}^T G^2(\underline{\theta})\underline{\omega} \\ \vdots \\ \underline{\omega}^T G^N(\underline{\theta})\underline{\omega} \end{vmatrix} \qquad (2.1)$$

where $\underline{\theta} = \{ \theta_1 \; \theta_2 \; \dots \; \theta_N \}^T$ the N-joint linear or angular position vector

$\underline{\omega} = \{ \omega_1 \; \omega_2 \; \dots \; \omega_N \}^T$ the N-joint linear or angular velocity vector

$\underline{\alpha} = \{ \alpha_1 \; \alpha_2 \; \dots \; \alpha_N \}^T$ the N-joint linear or angular acceleration vector

$\underline{T} = \{ T_1 \; T_2 \; \dots \; T_N \}^T$ the N-joint generalized driving force vector

$J_1(\underline{\theta},t)$ the N×N symmetric link inertia matrix

$\underline{l}(\underline{\theta},\underline{\omega},t)$ the vector of generalized external loads acting on the manipulator

$G^i(\underline{\theta})$ i= 1,2,...N the N×N symmetric matrix

containing centripetal and coriolis
torque weightings at each joint

The functions $J_1(\underline{\theta},t)$ and $G^i(\underline{\theta})$ are functions of the direction cosines of each joint coordinate and are therefore specific for each manipulator's geometry.

Each of the N linear permanent magnet DC motors is, for this development, assumed to be armature controlled. The differential equations for the N actuators may be written as

$$J_m\underline{\alpha} + \underline{T} = \bar{K}_T\underline{I} - A\underline{\theta} - B\underline{\omega} \tag{2.2}$$

$$\underline{I}' = -\bar{K}_V\underline{\omega} - \tilde{R}\underline{I} + L^{-1}\underline{V} \tag{2.3}$$

where $\underline{T}$ the N joint generalized loading torque vector (N-m)

$\underline{V} = \{ V_1\ V_2 \ldots V_N \}^T$ the N vector of input voltages to the armature (volts)

$\underline{I} = \{ I_1\ I_2 \ldots I_N \}^T$ the N vector of armature currents (amps)

$\underline{J_m}$ the N×N diagonal matrix of motor moments of inertia, relative to output shaft, $J_{mii} = J_{mi} \times r_i^2$ (N-m-sec^2)

r_i the gear box reduction ratio, $\frac{\underline{\omega_{in}}}{\omega_{out}}$ (rad/rad)

A the N×N diagonal matrix of actuator compliance coefficients (N-m/rad)

[illegible] the N×N diagonal matrix of actuator viscous friction coefficients (N-m/rad/sec)

$\bar{K}_T$ the N×N diagonal matrix of actuator torque constants at the output shaft by $\bar{K}_{Tii} = K_{Ti} \times r_i$

K_{Ti} motor torque constants (N-m/amp)

$\bar{R}$ a diagonal matrix of actuator constants from $R_{ii} = R_i / L_i$ (1/sec)

R_i actuator armature resistance (ohms)

L_i actuator armature inductance (henries)

$\bar{K}_v$ a diagonal matrix of actuator constants from $K_{vii} = K_{vi} \times r_i / L_i$

K_{vi} actuator armature back emf constants (volts/rad/sec)

The maximum rated actuator torques and speeds, and the maximum acceleration and speed are also assumed known.

Combining the differential equations for the manipulator links and the actuator dynamics results in the following augmented model which is based upon the definitions

$$\underline{X}_1 = \underline{\theta}$$
$$\underline{X}_2 = \underline{\omega} \qquad (2.4)$$
$$\underline{X}_3 = \underline{i}$$

The augmented system model after substitution is

$$\underline{X}_1' = \underline{X}_2 \tag{2.5}$$

$$\underline{X}_2' = [J_1(\underline{X}_1,t)+J_m]^{-1}\left(\tilde{K}_T\underline{X}_3 - 1(\underline{X}_1,\underline{X}_2,t) - \begin{vmatrix} X^T G^1(\underline{X})X \\ X^T G^2(\underline{X})X \\ \vdots \\ X^T G^N(\underline{X})X \end{vmatrix} - A\underline{X}_1 - B\underline{X}_2\right) \tag{2.6}$$

$$\underline{X}_3' = -\tilde{K}_V\underline{X}_2 - \bar{R}\underline{X}_3 + L^{-1}\underline{V} \tag{2.7}$$

This set of differential equations (2.5),(2.6),(2.7) form the complete state space model for the N-degree-of-freedom manipulator system with its actuators.

Linearizing and Decoupling Feedback

The uncompensated manipulator dynamic model with its actuators is a nonlinear coupled N-degree-of-freedom system. This system must be transformed by some feedback method in order for the manipulator system to meet the design specification of N linear decoupled subsystems for each link and its actuator. Following the method shown by Allon Guez, the transformation begins with a new linear state definition:

$$\begin{aligned} \underline{Y}_1 &= \underline{X}_1 \\ \underline{Y}_2 &= \underline{X}_2 \\ \underline{Y}_3 &= \underline{X}_2' \end{aligned} \tag{2.8}$$

Note that this represents the states of a third order system with the states of position, velocity, and acceleration.

Using the new state identity, the following differential equations may be defined:

$$\underline{Y}_1' = \underline{Y}_2 \tag{2.9}$$

$$\underline{Y}_2' = \underline{Y}_3 \tag{2.10}$$

$$\underline{Y}_3' = \underline{X}''_2 = \frac{d}{dt}\left([J_1(\underline{X}_1,t)+J_m]^{-1}\right)\left\{\tilde{K}_T\underline{X}_3-1(\underline{X}_1,\underline{X}_2,t)-\begin{vmatrix} X^TG^1(X)X \\ X^TG^2(\underline{X})X \\ \vdots \\ X^TG^N(\underline{X})X \end{vmatrix} -A\underline{X}_1 -B\underline{X}_2\right\} + (J_1(\underline{X}_1,t)+J_m)^{-1}\frac{d}{dt}\left[\tilde{K}_T\underline{X}_3-1(\underline{X}_1,\underline{X}_2,t) - \begin{vmatrix} X^TG^1(X)X \\ X^TG^2(\underline{X})X \\ \vdots \\ X^TG^N(\underline{X})X \end{vmatrix} -A\underline{X}_1 -B\underline{X}_2\right] \tag{2.11}$$

Henceforth for simplicity the following identities will be used:

$$G^* = \begin{vmatrix} X^TG^1(X)X \\ X^TG^2(\underline{X})X \\ \vdots \\ X^TG^N(\underline{X})X \end{vmatrix} \tag{2.12}$$

$$J^* = [J_1(\underline{X}_1,t)+J_M] \tag{2.13}$$

Thus, (2.11) can be expressed as

$$\underline{Y}_3' = \underline{X}''_2 = \frac{d}{dt}[J^{*-1}]\{\tilde{K}_T\underline{X}_3 -1(\underline{X}_1,\underline{X}_2,t) -G^* -A\underline{X}_1 -B\underline{X}_2\} + J^{*-1}\frac{d}{dt}[\tilde{K}_T\underline{X}_3 -1(\underline{X}_1,\underline{X}_2,t) -G^* -A\underline{X}_1 -B\underline{X}_2] \tag{2.14}$$

Further, in order to introduce an equation containing the feedback voltage $\underline{V}$,

$$\underline{Y}_3' = \frac{d}{dt}[J^{*-1}]\{\bar{K}_T\underline{X}_3 - l(\underline{X}_1,\underline{X}_2,t) - G^* - A\underline{X}_1 - B\underline{X}_2\} + J^{*-1}\bar{K}_T\underline{X}_3'$$

$$+ J^{*-1}\frac{d}{dt}[-l(\underline{X}_1,\underline{X}_2,t) - G^* - A\underline{X}_1 - B\underline{X}_2] \qquad (2.15)$$

Substituting equation (2.7) into (2.15) one obtains

$$\underline{Y}_3' = \frac{d}{dt}[J^{*-1}]\{\bar{K}_T\underline{X}_3 - l(\underline{X}_1,\underline{X}_2,t) - G^* - A\underline{X}_1 - B\underline{X}_2\}$$

$$+ J^{*-1}\bar{K}_T\{-\bar{K}_V\underline{X}_2 - \bar{R}\underline{X}_3 + L^{-1}\underline{V}\}$$

$$+ J^{*-1}\frac{d}{dt}[-l(\underline{X}_1,\underline{X}_2,t) - G^* - A\underline{X}_1 - B\underline{X}_2] \qquad (2.16)$$

In order to complete the transformation to a triple integrator system, the following differential equation must be realized:

$$\underline{Y}_3' = \underline{Z} \qquad (2.17)$$

where $\underline{Z}$ is the new jerk N vector input to the actuators. Note that the armature input voltage may be solved in equation (2.16) for that feedback voltage $\underline{V}$ that will enable the triple integrator system to be formed. The solution for the nonlinear voltage $\underline{V}$ is

$$\underline{V}(\underline{X},\underline{Z},t) = \tilde{K}_T^{-1} J^*(\underline{X}_1,t)[\underline{Z} - \frac{d}{dt} J^{*-1}\{\tilde{K}_T\underline{X}_3 - \underline{l}(\underline{X}_1,\underline{X}_2,t) - G^* - A\underline{X}_1 - B\underline{X}_2\}]$$

$$+ \tilde{K}_T^{-1} \frac{d}{dt} [\ \underline{l}(\underline{X}_1,\underline{X}_2,t) + G^* + A\underline{X}_1 + B\underline{X}_2\ ]$$

$$+ L\tilde{K}_T^{-1}[\ \bar{K}_T\bar{K}_V\underline{X}_2 + \bar{K}_T\bar{R}\underline{X}_3\ ] \qquad (2.18)$$

The feedback voltage $\underline{V}(\underline{X},\underline{Z},t)$ may be further transformed because the state of motor current $\underline{X}_3$ does not appear in the new state definition (2.1). There is an advantage in the elimination of the motor current as a feedback variable when the need to evaluate the derivative of the inverse of $J^*(\underline{X},t)$ occurs. Recall that

$$\underline{Y}_3 = \underline{X}_2' = J^{*-1}(\underline{X}_1,t)(\bar{K}_T\underline{X}_3 - \underline{l}(\underline{X}_1,\underline{X}_2,t) - G^* - A\underline{X}_1 - B\underline{X}_2) \qquad (2.19)$$

Solving for the product $\tilde{K}_T\underline{X}_3$ and using the new state identities

$$\tilde{K}_T\underline{X}_3 = A\underline{Y}_1 + B\underline{Y}_2 + G^*(\underline{Y}_1,\underline{Y}_2) + \underline{l}(\underline{Y}_1,\underline{Y}_2,t) + J^*(\underline{Y}_1,t)\underline{Y}_3 \qquad (2.20)$$

and thus

$$\underline{X}_3 = \bar{K}_T^{-1}[A\underline{Y}_1 + B\underline{Y}_2 + G^*(\underline{Y}_1,\underline{Y}_2) + \underline{l}(\underline{Y}_1,\underline{Y}_2,t) + J^*(\underline{Y}_1,t)\underline{Y}_3] \qquad (2.21)$$

Substituting (2.20) and (2.21) into (2.18) one obtains

$$\underline{V}(\underline{Y},\underline{Z},t) = L\bar{K}_T^{-1}J^*(\underline{Y}_1,t)\{\ \underline{Z} - \frac{d}{dt}[J^{*-1}\text{[illegible]}(\underline{Y}_1,t)\underline{Y}_3]$$

$$-L\bar{K}_T^{-1}\frac{d}{dt}[\ \underline{1}(\underline{Y}_1,\underline{Y}_2,t) + G^*(\underline{Y}_1,\underline{Y}_2) + A\underline{Y}_1 + B\underline{Y}_2]$$

$$+L\bar{K}_V\underline{Y}_2 + L\bar{R}\bar{K}_T^{-1}[A\underline{Y}_1 + B\underline{Y}_2 + G^*(\underline{Y}_1,\underline{Y}_2) + \underline{1}(\underline{Y}_1,\underline{Y}_2,t)$$

$$J^*(\underline{Y}_1,t)\underline{Y}_3]\ \} \tag{2.22}$$

Using the fact that the inverse of a diagonal matrix is diagonal and that diagonal matrices commute, then $\bar{R}\bar{K}_T^{-1} = \bar{K}_T^{-1}\bar{R}$. For a nonsingular square matrix such as J^* the following holds

$$\frac{d}{dt}[J^{*-1}(\underline{Y},t)] = -J^{*-1}(\underline{Y}_1,t)\,\frac{d}{dt}[J^*(\underline{Y}_1,t)]\,J^{*-1}(\underline{Y}_1,t) \tag{2.23}$$

With these definitions the linearizing and decoupling feedback becomes

$$\underline{V}(\underline{Y},\underline{Z},t) = L\bar{K}_T^{-1}\{\ J^*(\underline{Y}_1,t)\underline{Z}$$

$$+J^*(\underline{Y},t)J^{*-1}(\underline{Y}_1,t)\frac{d}{dt}[J^*(\underline{Y}_1,t)]J^{*-1}(\underline{Y}_1,t)J^*(\underline{Y}_1,t)\underline{Y}_3$$

$$+\frac{d}{dt}[\ \underline{1}(\underline{Y}_1,\underline{Y}_2,t) + G^*(\underline{Y}_1,\underline{Y}_2)\ ] \qquad + A\underline{Y}_2 + B\underline{Y}_3 + \bar{K}_T\bar{K}_V\underline{Y}_2$$

$$+\bar{R}[\ A\underline{Y}_1 + B\underline{Y}_2 + G^*(\underline{Y}_1,\underline{Y}_2) + \underline{1}(\underline{Y}_1,\underline{Y}_2,t) + J^*(\underline{Y}_1,t)\underline{Y}_3]\ \} \tag{2.24}$$

Final simplification and grouping of gains associated with the system states results in the voltage necessary to realize the linear decoupled manipulator system

$$
\begin{aligned}
\underline{V}(\underline{Y},\underline{Z},t) = \tilde{L}\tilde{K}_T^{-1}\{\ & J^*(\underline{Y}_1,t)\underline{Z} + \left(\frac{d}{dt}[J^*(\underline{Y}_1,t)] + \tilde{R}J^*(\underline{Y}_1,t) + B\right)\underline{Y}_3 \\
& + \frac{d}{dt}[\ l(\underline{Y}_1,\underline{Y}_2,t) + G^*(\underline{Y}_1,\underline{Y}_2)\ ] \\
& + (\ \tilde{K}_T\tilde{K}_V + \tilde{R}B + A\ )\underline{Y}_2 + (\ \tilde{R}A\ )\underline{Y}_1 \\
& + \tilde{R}(\ G^*(\underline{Y}_1,\underline{Y}_2) + \underline{l}(\underline{Y}_1,\underline{Y}_2,t)\ )\ \}
\end{aligned}
\qquad (2.25)
$$

This feedback voltage vector is assumed to be applied to any actuator and manipulator system used in the remainder of this dissertation. Consequently, each degree-of-freedom will appear to perform independently as a linear triple integrator system.

CHAPTER III
TIME OPTIMAL POSITION CONTROL (TOPC)

The manipulator system now appears as N linear, third order subsystems by the use of the feedback linearization and decoupling transformation. Each subsystem has actuator constraints that limit the obtainable speed, acceleration and jerk. For this chapter it is assumed that these limits are expressed as known values of allowable jerk input A_i, maximum allowable acceleration α_{m_i}, and maximum allowable velocity ω_{m_i}. In terms of our transformed manipulator states these constraints are

$$|\, \underline{Y}_{2_i} \,| \le \omega_{m_i} \tag{3.1}$$

$$|\, \underline{Y}_{3_i} \,| \le \alpha_{m_i} \tag{3.2}$$

$$|\, \dot{\underline{Y}}_{3_i} \,| \le A_i \tag{3.3}$$

Note that ω_{m_i} and α_{m_i} are in fact constraints on the state variables of each subsystem, and A_i is the limited control input to each subsystem. As developed in "Optimal Control of Robotic Manipulators" by Allon Guez [10,1983], the simplest problem of minimum time control is that of moving from a known static position to another known static position in minimum time. A very short move of each joint of the manipulator could be performed without reaching any state

variable constraints. This, then, was the first solution given by Guez as the unconstrained case.

Case 1 Unconstrained Motion

The state equations which the optimal control must satisfy are those of a linear, time invariant triple integrator system formed by the PLOT. They are

$$\underline{Y}_1' = \underline{Y}_2 \tag{3.4}$$

$$\underline{Y}_2' = \underline{Y}_3 \tag{3.5}$$

$$\underline{Y}_3' = \underline{Z} \tag{3.6}$$

These state equations have some initial conditions

$$\underline{Y}_1(0) = \underline{Y}_{1_0} \tag{3.7}$$

$$\underline{Y}_2(0) = \underline{Y}_{2_0} = \underline{0} \tag{3.8}$$

$$\underline{Y}_3(0) = \underline{Y}_{3_0} = \underline{0} \tag{3.9}$$

that, together with the state equations, must be satisfied in a trajectory to the non-empty terminal set Θ of R^n. The terminal set in the static endpoint cases is assumed to be

$$\Theta = [\underline{0}], \qquad \Theta \in R^{3 \times n} \tag{3.10}$$

The control constraint set $\underline{U}$ is the finite jerk input to each actuator:

$$\underline{Z} \in \underline{U}, \qquad \underline{U} \in R^n \tag{3-11}$$

The cost associated with minimum time optimal control is the minimization of the function

$$V[\underline{Y}_0,\underline{U},\underline{Y}] = \int_{t_0}^{t_f} f^0(\underline{Y},\underline{U})dt, \text{ where } f^0(\underline{Y},\underline{U}) = 1 \qquad (3.13)$$

Therefore the function to minimize is

$$V[\underline{Y}_0,\underline{U},\underline{Y}] = t_f - t_0 \qquad (3.14)$$

Normally the solution of such an optimal control problem satisfying some initial and final conditions would be carried out by some 'backing out' procedure that integrates the adjoint variables back in time from the terminal state. The adjoint variables are a function of the state equations [15,Leitmann,1981] and define the time when an optimal control input would be applied. The solution of such optimal control equations is usually nonlinear and requires iteration; further, for many problems, solutions do not exist. However, due to the linear augmented system $\underline{Y}$ and the static endpoint conditions, the solution to the optimal control problem can be shown to be expressed as a closed form solution to a switching surface F, [10,Gumz,1983,p66], [25,Smith,1966],

$$F = Y_1 + Y_3^3/3 + \epsilon Y_2 Y_3 + \epsilon[Y_3^2/2 - \epsilon Y_2]^{3/2} \qquad (3.15)$$

where

$$\varepsilon = \mathrm{sgn}(Y_2 + Y_3 \ldots) \qquad (3.16)$$

The optimal input $\hat{Z}(t)$ can be expressed as

$$\hat{Z}(t) = \begin{cases} \Delta & 0 \le t < \frac{t_f}{4} \\ -\Delta & \frac{t_f}{4} \le t < \frac{3t_f}{4} \\ \Delta & \frac{3t_f}{4} \le t \le t_f \end{cases} \qquad (3.17)$$

where for total initial displacement from target (origin) ξ

$$t_f = 2\left(\frac{4\xi}{\Delta}\right)^{1/3} \qquad (3.18)$$

Guez then integrated the jerk inputs to find the solution to acceleration, velocity, and position as per the linear state equations. These solutions are

$$\hat{Y}_3(t) = \begin{cases} \Delta t & 0 \le t < \frac{t_f}{4} \\ \Delta(\frac{t_f}{2}-t) & \frac{t_f}{4} \le t < \frac{3t_f}{4} \\ \Delta(t-t_f) & \frac{3t_f}{4} \le t \le t_f \end{cases} \qquad (3.19)$$

$$\hat{Y}_2(t) = \begin{cases} \frac{\Delta t^2}{2} & 0 \le t < \frac{t_f}{4} \\ \frac{\Delta t_f^2}{16} - \frac{\Delta}{2}\left(t - \frac{t_f}{2}\right)^2 & \frac{t_f}{4} \le t < \frac{3t_f}{4} \\ \frac{\Delta}{2}(t - t_f)^2 & \frac{3t_f}{4} \le t \le t_f \end{cases} \tag{3.20}$$

$$\hat{Y}_1(t) = \begin{cases} -\zeta + \frac{\Delta t^3}{6} & 0 \le t < \frac{t_f}{4} \\ -\zeta + \frac{\Delta t_f^3}{192} - \frac{\Delta t_f^2 t}{16} + \frac{\Delta t_f t^2}{4} - \frac{\Delta t^3}{6} & \frac{t_f}{4} \le t < \frac{3t_f}{4} \\ -\zeta - \frac{26\Delta t_f^3}{192} - \frac{\Delta t_f^2 t}{2} - \frac{\Delta t_f t^2}{2} + \frac{\Delta t^3}{6} & \frac{3t_f}{4} \le t \le t_f \end{cases} \tag{3.21}$$

The constrained cases for minimum time moves of the triple integrator system would occur as the required positional moves became long enough so that the limits of maximum acceleration and velocity were reached. The crossover point for the different cases occurs as a function of the required move and the maximum attainable acceleration and velocity. The first case (unconstrained motion) ends with a move that reaches the maximum acceleration from equation (3.19) as

$$a_m = \frac{\Delta t_f}{4} = \frac{\Delta}{2}\left(\frac{4\zeta}{\Delta}\right)^{1/3} \tag{3.22}$$

Solving for ζ at the maximum

$$t < \frac{\Delta}{4}\left(\frac{2a_m}{\Delta}\right)^3 \qquad (3.23)$$

$$\zeta < \frac{2a_m^3}{\Delta^2} \qquad (3.24)$$

Equation (3.24) sets the upper bound of case 1 for the unconstrained trajectory move.

Case 2 Constrained Acceleration

Guez used the work of Smith [25,1966] to include the case of constrained acceleration during a portion of the trajectory. The solutions of the switching surface for the optimal input $\hat{Z}(t)$ are given by the following

$$\hat{Z}(t) = \begin{cases} \Delta & 0 \le t < \frac{a_m}{\Delta} \\ 0 & \frac{a_m}{\Delta} \le t < \frac{t_f}{2} - \frac{a_m}{\Delta} \\ -\Delta & \frac{t_f}{2} - \frac{a_m}{\Delta} \le t < \frac{t_f}{2} + \frac{a_m}{\Delta} \\ 0 & \frac{t_f}{2} + \frac{a_m}{\Delta} \le t < t_f - \frac{a_m}{\Delta} \\ \Delta & t_f - \frac{a_m}{\Delta} \le t < t_f \end{cases} \qquad (3.25)$$

The corresponding functions for acceleration, velocity and position (during the applied optimal input $\hat{Z}(t)$) are

$$
\hat{Y}_3(t) = \begin{cases} \Delta t & 0 \le t < \frac{a_m}{\Delta} \\ a_m & \frac{a_m}{\Delta} \le t < \frac{t_f}{2} - \frac{a_m}{\Delta} \\ \frac{\Delta}{2}(t_f - \Delta t) & \frac{t_f}{2} - \frac{a_m}{\Delta} \le t < \frac{t_f}{2} + \frac{a_m}{\Delta} \\ -a_m & \frac{t_f}{2} + \frac{a_m}{\Delta} \le t < t_f - \frac{a_m}{\Delta} \\ \Delta(t - t_f) & t_f - \frac{a_m}{\Delta} \le t < t_f \end{cases} \tag{3.26}
$$

$$
\hat{Y}_2(t) = \begin{cases} \frac{\Delta t^2}{2} & 0 \le t < \frac{a_m}{\Delta} \\ a_m t - \frac{a_m^2}{2\Delta} & \frac{a_m}{\Delta} \le t < \frac{t_f}{2} - \frac{a_m}{\Delta} \\ \Delta\left[\frac{t_f}{2}t - \frac{t^2}{2} - \frac{t_f^2}{8} - \left(\frac{a_m}{\Delta}\right)^2 + \frac{a_m}{2\Delta}t_f\right] & \frac{t_f}{2} - \frac{a_m}{\Delta} \le t < \frac{t_f}{2} + \frac{a_m}{\Delta} \\ a_m t_f - \frac{a_m^2}{2\Delta} & \frac{t_f}{2} + \frac{a_m}{\Delta} \le t < t_f - \frac{a_m}{\Delta} \\ \frac{\Delta}{2}(t - t_f)^2 & t_f - \frac{a_m}{\Delta} \le t < t_f \end{cases} \tag{3.27}
$$

$$
\hat{Y}_1(t) = \begin{cases}
-\zeta + \frac{\Delta t^3}{6} & 0 \le t < \frac{a_m}{\Delta} \\
-\zeta - \frac{a_m^2 t}{2\Delta} + \frac{a_m t^2}{2} + \frac{a_m^3}{6\Delta^2} & \frac{a_m}{\Delta} \le t < \frac{t_f}{2} - \frac{a_m}{\Delta} \\
-\zeta + \Delta[\frac{t_f t^2}{4} - \frac{t^3}{6} + \frac{t_f^3}{48} - \frac{t_f^2 t}{8}] & \frac{t_f}{2} - \frac{a_m}{\Delta} \le t < \frac{t_f}{2} + \frac{a_m}{\Delta} \\
\quad + a_m[\frac{t_f}{2}t - \frac{a_m t}{\Delta} - \frac{t_f^2}{8} + \frac{a_m t_f}{4\Delta}] & \\
-\zeta + a_m[t_f t - \frac{t^2}{2} - \frac{a_m t}{2\Delta} - \frac{a_m^2}{6\Delta^2} - \frac{a_m t_f^2}{4}] & \frac{t_f}{2} + \frac{a_m}{\Delta} \le t < t_f - \frac{a_m}{\Delta} \\
-\zeta + \Delta[\frac{t^3}{6} - \frac{t_f t^2}{2} + \frac{t_f^2 t}{2} - \frac{t_f^3}{6}] & t_f - \frac{a_m}{\Delta} \le t < t_f \\
\quad + a_m[\frac{t_f^2}{4} - \frac{t_f a_m}{2\Delta}] &
\end{cases}
\tag{3.28}
$$

where t_f is the solution to $\hat{Y}_1(t_f) = 0$ in equation (3.28),

$$
t_f = \frac{a_m}{\Delta} + \sqrt{\frac{a_m}{\Delta} + \frac{4\zeta}{a_m}} \quad . \tag{3.29}
$$

Observing that the maximum velocity occurs at the midpoint for a constrained acceleration move, the solution to the upper bound for this case's velocity can be found by

$$
\omega_m > Y_{2_{max}} \tag{3.30}
$$

with

$$
Y_{2_{max}} = \hat{Y}_2 \Big|_{t=\frac{t_f}{2}} = \sqrt{\frac{a_m^2}{4\Delta^2} + a_m \zeta} - \frac{a_m^2}{2\Delta} \tag{3.31}
$$

Case 3 Constrained Acceleration and Velocity

The case of constrained acceleration and constrained velocity would occur when ω_m was less than that required in a long position trajectory as

$$\omega_m \leq \sqrt{\frac{a_m^2}{4A^2} + a_m \times \zeta} - \frac{a_m^2}{2A} \tag{3.32}$$

The optimal input $\hat{Z}(t)$ and the resulting optimal trajectory states are given as

$$\hat{Z}(t) = \begin{cases} A & 0 \leq t < \frac{a_m}{A} \\ 0 & \frac{a_m}{A} \leq t < \frac{\omega_m}{a_m} \\ -A & \frac{\omega_m}{a_m} \leq t < \frac{a_m}{A} + \frac{\omega_m}{a_m} \\ 0 & \frac{a_m}{A} + \frac{\omega_m}{a_m} \leq t < \frac{\zeta}{\omega_m} \\ -A & \frac{\zeta}{\omega_m} \leq t < \frac{\zeta}{\omega_m} + \frac{a_m}{A} \\ 0 & \frac{\zeta}{\omega_m} + \frac{a_m}{A} \leq t < \frac{\zeta}{\omega_m} + \frac{\omega_m}{a_m} \\ A & \frac{\zeta}{\omega_m} + \frac{\omega_m}{a_m} \leq t < t_f \end{cases} \tag{3.33}$$

where

$$t_f = \frac{\zeta}{\omega_m} + \frac{a_m}{A} + \frac{\omega_m}{a_m} \tag{3.34}$$

$$
\bar{\alpha}(t) = \begin{cases}
\Delta t & 0 \le t < \frac{\alpha_m}{\Delta} \\
\alpha_m & \frac{\alpha_m}{\Delta} \le t < \frac{\omega_m}{\alpha_m} \\
-\Delta\left(t - \frac{\alpha_m}{\Delta} - \frac{\omega_m}{\alpha_m}\right) & \frac{\omega_m}{\alpha_m} \le t < \frac{\alpha_m}{\Delta} + \frac{\omega_m}{\alpha_m} \\
0 & \frac{\alpha_m}{\Delta} + \frac{\omega_m}{\alpha_m} \le t < \frac{\xi}{\omega_m} \\
-\Delta\left(t - \frac{\xi}{\omega_m}\right) & \frac{\xi}{\omega_m} \le t < \frac{\xi}{\omega_m} + \frac{\alpha_m}{\Delta} \\
-\alpha_m & \frac{\xi}{\omega_m} + \frac{\alpha_m}{\Delta} \le t < \frac{\xi}{\omega_m} + \frac{\omega_m}{\alpha_m} \\
\Delta(t - t_f) & \frac{\xi}{\omega_m} + \frac{\omega_m}{\alpha_m} \le t < t_f
\end{cases} \qquad (3.35)
$$

$$
\bar{\omega}(t) = \begin{cases}
\frac{\Delta t^2}{2} & 0 \le t < \frac{\alpha_m}{\Delta} \\
\alpha_m\left(t - \frac{\alpha_m}{2\Delta}\right) & \frac{\alpha_m}{\Delta} \le t < \frac{\omega_m}{\alpha_m} \\
\omega_m - \frac{\Delta}{2}\left(t - \frac{\alpha_m}{\Delta} - \frac{\omega_m}{\alpha_m}\right)^2 & \frac{\omega_m}{\alpha_m} \le t < \frac{\alpha_m}{\Delta} + \frac{\omega_m}{\alpha_m} \\
\omega_m & \frac{\alpha_m}{\Delta} + \frac{\omega_m}{\alpha_m} \le t < \frac{\xi}{\omega_m} \\
\omega_m - \frac{\Delta}{2}\left(t - \frac{\xi}{\omega_m}\right)^2 & \frac{\xi}{\omega_m} \le t < \frac{\xi}{\omega_m} + \frac{\alpha_m}{\Delta} \\
\omega_m - \alpha_m\left(t - \frac{\xi}{\omega_m} - \frac{\alpha_m}{2\Delta}\right) & \frac{\xi}{\omega_m} + \frac{\alpha_m}{\Delta} \le t < \frac{\xi}{\omega_m} + \frac{\omega_m}{\alpha_m} \\
\frac{\Delta}{2}(t - t_f)^2 & \frac{\xi}{\omega_m} + \frac{\omega_m}{\alpha_m} \le t < t_f
\end{cases} \qquad (3.36)
$$

$$
\hat{Y}_1(t) = \begin{cases}
-\zeta + \frac{\Delta t^3}{6} & 0 \le t < \frac{a_m}{\Delta} \\[4pt]
-\zeta + \frac{a_m}{2}\left(t - \frac{a_m}{2\Delta}\right)^2 + \frac{a_m^3}{24\Delta^2} & \frac{a_m}{\Delta} \le t < \frac{\omega_m}{a_m} \\[4pt]
-\zeta - \frac{\Delta}{6}\left(t - \frac{a_m}{\Delta} - \frac{\omega_m}{a_m}\right)^3 & \frac{\omega_m}{a_m} \le t < \frac{a_m}{\Delta} + \frac{\omega_m}{a_m} \\
\quad + \omega_m\left(t - \frac{\omega_m}{2a_m} - \frac{a_m}{2\Delta}\right) & \\[4pt]
-\zeta + \omega_m\left(t - \frac{\omega_m}{2a_m} - \frac{a_m}{2\Delta}\right) & \frac{a_m}{\Delta} + \frac{\omega_m}{a_m} \le t < \frac{\zeta}{\omega_m} \\[4pt]
-\zeta + \omega_m\left(t - \frac{\omega_m}{2a_m} - \frac{a_m}{2\Delta}\right) & \frac{\zeta}{\omega_m} \le t < \frac{\zeta}{\omega_m} + \frac{a_m}{\Delta} \\
\quad - \frac{\Delta}{6}\left(t - \frac{\zeta}{\omega_m}\right)^3 & \\[4pt]
-\zeta + \omega_m\left(t - \frac{\omega_m}{2a_m} - \frac{a_m}{2\Delta}\right) & \frac{\zeta}{\omega_m} + \frac{a_m}{\Delta} \le t < \frac{\zeta}{\omega_m} + \frac{\omega_m}{a_m} \\
\quad - \frac{a_m}{2}\left(t - \frac{\zeta}{\omega_m} - \frac{a_m}{2\Delta}\right)^2 - \frac{a_m^3}{24\Delta^2} & \\[4pt]
\frac{\Delta}{6}\left(t - \frac{\zeta}{\omega_m} - \frac{a_m}{\Delta} - \frac{\omega_m}{a_m}\right)^3 & \frac{\zeta}{\omega_m} + \frac{\omega_m}{a_m} \le t < t_f
\end{cases}
\tag{3.37}
$$

Trajectories resulting from any of these cases (1, 2, or 3) will be called "nominal trajectories" for the remainder of the dissertation. That is, these trajectories are solutions of minimum time motion trajectories from static initial and final endpoints for a linear time invariant triple integrator system. All of the cases have solutions based upon the observable fact for electro-servomechanisms the system will always reach its maximum (minimum) acceleration before achieving its maximum (minimum) velocity. Gust

[10,1983] made this assumption based upon the electrical time constant of the actuators being two orders of magnitude smaller than the mechanical time constant of the actuator and load inertia. Therefore the actuator reaches its maximum current, which is related to torque and thus acceleration, before reaching its maximum voltage, which is related to the applied jerk.

A method was sought to obtain an optimal solution of switching times for the constrained input $\hat{Z}(t)$ for systems that, while on a "nominal trajectory", were reassigned the desired final state of the manipulator. This situation would be consistent with that of the intelligent controller encountering new conditions. The next chapter outlines the method used to find such optimal inputs for the reassignment cases. The knowledge obtained from this chapter in how the system states are performing under specific optimum jerk inputs which occur in a specific order will be of great benefit to finding the new solutions.

CHAPTER IV
TOPC WITH MIDSTREAM TRAJECTORY REASSIGNMENT

Systems with non-zero initial state derivatives must have new input switching times derived that define a new optimal trajectory. These non-static endpoint cases may result from reassignment of the final goal during a nominal trajectory. The method of finding these new switching times and the resulting trajectory will follow in this section. The method used adheres to a set of rules that do not violate the state constraints and allow the concatenating of trajectories together from portions of similar optimal moves.

Underlying Assumptions

The approach to a solution of minimal time control with midstream reassignment will be one that operates with several assumptions. These assumptions set the conditions under which an intelligent controller can properly generate a minimum time trajectory. First, the method of linearizing and decoupling the system to be controlled is assumed to be currently in operation. Therefore, all inputs will be jerks (rad/sec^3) applied to time invariant linear third order decoupled subsystems. The reassignment is a new final static position for each subsystem. These reassignments are assumed appropriate for the geometry and configuration

limits of the specific manipulator. Thus no transformation of world coordinates is to be performed at this level of control as the subsystems will operate in a decoupled state variable environment.

Another assumption is that an analytical solution for minimum time inputs with state variable and input constraints for nominal trajectories exists (see chapter II) and an optimal input has been or is being applied to a nominal trajectory. The current nominal trajectory is a function of known system constraints of maximum jerk input and maximum system acceleration and velocity. Further, the current status of the system is assumed to be only a function of the applied input and of the elapsed time of a nominal trajectory from the initial starting position. The reassignments are therefore assumed to be concatenated with nominal minimum time trajectories or other reassignment inputs for any trajectory targets. The concatenation of reassigned inputs with those of a nominal trajectory is to be performed such that any subsequent reassignment will be allowed.

Fundamental to the generation of such reassignment of inputs is the fact that the control input jerks, $\hat{z}_i(t)$, may be joined together as

$$\hat{z}_i(t) = \hat{u}_i(t) \qquad \text{for } t \in [t_0, t^*], \tag{4.1}$$

$$\hat{z}_1(t) = \overset{*}{u}_1(t) \qquad \text{for } t\in[t^*, t_f] \qquad (4.2)$$

where $t_0 \le t^* \le t_f$ and

$$\hat{u}_1(t)\epsilon U, \text{ and } \overset{*}{u}_1(t)\epsilon U. \qquad (4.3)$$

Thus $\hat{z}_1(t)$ is an admissible control since is satisfies the condition that $\hat{z}_1(t)$ is piecewise continuous and $u(t)\epsilon U$, $\forall[t_0,t_f]$ [15,Leitmann,1981,p83]. Also the control defined by the joining of $\hat{u}_1(t)$ and $\overset{*}{u}_1(t)$ generates the solution

$$\underline{Y}(t) = \underline{Y}(t)\ , \qquad t\epsilon[t_0,t^*], \qquad (4.4)$$

$$\underline{Y}(t) = \underline{Y}^*(t), \qquad t\epsilon[t^*,t_f]. \qquad (4.5)$$

Therefore the cost corresponding to these inputs is

$$V[\underline{Y}_0,\underline{U},\underline{Y}] = \int_{t_0}^{t^*} f^0(\underline{Y},\underline{U})dt \qquad (4.6)$$

$$V[\underline{Y}*,\underline{U}^*,\underline{Y}^*] = \int_{t^*}^{t_f} f^0(\underline{Y}^*,\underline{U}^*)dt \qquad (4.7)$$

$$V[\underline{Y}_0,\bar{U},\bar{Y}] = \int_{t_0}^{t_f} f^0(\bar{Y},\bar{U})dt. \qquad (4.8)$$

Then

$$V[\underline{Y}_0,\underline{U},\underline{Y}] + V[\underline{Y}*,\underline{U}^*,\underline{Y}^*] = \qquad V[\underline{Y}_0,\bar{U},\bar{Y}] \qquad (4.9)$$

and the total cost along a solution $\tilde{Y}$ is the sum of the cost along $\underline{Y}(t)$ and $\underline{Y}^*(t)$, that is, the cost is additive [Leitmann,1981]. A logical consequence of this additive cost is that the cost of performing a nominal trajectory (the time elapsed is the cost) cannot be recovered. Thus the best the controller can do if a reassignment occurs is to perform from the current state and time forward into the future.

Real Time Decisions

The task of midstream trajectory reassignment contains two separate functions that are best performed with the knowledge that both will exist even though they may be performed at different times. The two functions are (1) the real time decisions of what type of input to correctly apply to the triple integrator systems and (2), how to build the switch times for the jerk inputs. The rule based real time decisions retrieve a set of switch times from the correct table of switch times. The manner in which switch times are calculated depends upon how the controller will access these proper switch times after determining which is the necessary category of switch times in which to search.

Determining the table in which to search for the jerk input switch times occurs after an analysis of the current state of the dynamic system under control. That is, on what type of move is the degree-of-freedom? Is the current

motion one that has required maximum acceleration, or maximum acceleration and maximum velocity, or is the motion unrestrained? These three types of moves are those that were constructed in chapter III for time optimal position control. The positioning of each system corresponds to the cases 1, 2, or 3. The first decision level, which reassignment input to provide, will be based on what case is currently active by:

case 1 unconstrained motion (4.10)

case 2 constrained acceleration

case 3 constrained acceleration and velocity

The next level of decision is based upon when the reassignment occurs, i.e., when (relative to which portion of a nominal move) the controller is made aware of the reassignment. The nominal moves from chapter III can be divided into seven general sectors, each with a unique combination of applied jerk, resulting acceleration, velocity and displacement. For the shorter moves (shorter being defined by the actuator constraints) in cases 1 and 2, some sectors have no time duration; that is, all moves have a common base of seven applied jerk durations. In addition, all the nominal moves in chapter III have the same jerk orientation at each sector (for the same direction move), whether positive jerk, no jerk, or negative jerk. The seven sectors are outlined below with their characteristic state of acceleration and velocity.

Sector 1. Sector 1 has applied positive jerk with the result of linearly increasing acceleration and increasing parabolic velocity. If on a case 1 move, the maximum acceleration will not be reached, nor will the maximum velocity. If the system is on a case 2 or 3 trajectory the acceleration will be at maximum at the end of this sector.

Sector 2. Sector 2 has no jerk input applied to the system, therefore there can be no increase in acceleration. If a case 1 move is active this sector exists for zero (0) time. For a case 2 or 3 trajectory, the system has a linearly increasing velocity with slope of maximum acceleration a_m.

Sector 3. During sector 3, the jerk is negative in order to reduce the acceleration of the system back to zero and consequently slow the rate of change of velocity. If a case 3 move is active the system will be at maximum velocity at the end of this sector.

Sector 4. This sector exists (is not of zero duration) only during a case 3 move which requires some finite non zero dwell at constant velocity. There is no input jerk applied during this sector and no acceleration since the acceleration was reduced to zero in sector 3.

Sector 5. Sector 5 begins the negative acceleration or slowing down of the system with an applied negative jerk input. The negative applied jerk effects a linearly increasing negative acceleration and a parabolic reduction

in the velocity. A case 2 or 3 motion will result in the maximum negative acceleration being applied at the end of this sector.

Sector 6. As in sector 2, the maximum acceleration is active in this sector (opposite in sign from sector 2). The velocity is decreasing with slope of $-a_m$ in this sector with no applied jerk.

Sector 7. The last sector has a positive jerk applied in order to reduce the maximum negative acceleration to zero. The velocity is consequently parabolically reduced to zero as the system slows to a stop. The degree-of-freedom is at rest at the end of this sector for all cases 1, 2, or 3 with no acceleration of velocity.

The final decision level is used to determine which reassignment inputs to apply to the actuator. The choice of which reassignment to apply is based upon the relative difference between the reassignment target position and the current nominal target position. This relative difference can be broken down into three separate moves

1) requiring the system to move further forward than the original nominal move,
2) stopping short of the original nominal move and,
3) requiring a reversal of the direction with resulting (relative) negative velocity

The ability of the degree-of-freedom to perform these

three moves is specific to each case and sector that currently exist.

The decision as to whether the required reassignment will fall into the category of continuing forward, stopping short, or requiring a backward motion is not an intuitively obvious choice for many reassignment possibilities. A trivial move would be one in which the choice is obvious. For example, a reassignment target that required continuing further forward when the reassignment occurred in case 3, sector 1, 2, 3 or before the end of sector 4 is obvious because the system must continue on in the constant velocity sector 4. The extension of case 3, sector 4 would be the area of a rectangle whose width is the extension time and whose height is the maximum velocity ω_m. The solution to this decision would be one that produces an extended constant velocity of duration in sector 4.

$$t_4 = \frac{\xi_r - \xi_n}{\omega_m} \tag{4.11}$$

where

$$\xi_r = \text{reassignment target (rad)} \tag{4.12}$$

$$\xi_n = \text{nominal target (rad)} \tag{4.13}$$

A nonobvious move, for example, would occur when the reassignment demand came during a case 3, sector 5 or 6 motion which required a final position that is less than

that of the nominal move. The system cannot slow down any faster during sectors 5 or 6 since the maximum negative acceleration is currently being applied. Thus the reassignment motion will reverse the direction of the nominal move and back up to reach the new position. It should be noted that to come to a complete stop on the nominal move and then reverse the direction is a non-optimal motion. This reassignment will be shown in detail in chapter V.

The rules for the real time decision can be based upon knowledge of the system's current state. Knowledge of the active case (1, 2, or 3) is used first, next, knowledge of the active sector (1,2,...7) in which the reassignment has occurred, finally, the knowledge of the required move of the nominal target position relative to the reassignment target position. These decision rules are invariant from joint controller to joint controller, which offers a high degree of desirable replication in the controller design, reducing the development and manufacturing cost. The parameters relative to the actual decision making are the time duration of the current nominal move (a system real time clock) and the a priori known maximum input jerk, maximum allowable acceleration and maximum allowable velocity for each actuator system.

Reassignment Trajectory Construction

The construction of minimum time trajectories that must be concatenated with those of nominal trajectories is

performed with rules of the concatenation in mind. Since the real time decisions of which reassignment trajectory to use will be based upon the unique case of past, present and future motion, the generation of trajectories must follow the same rules. The past knowledge is what specific case (1, 2, or 3) was chosen for the nominal trajectory, the dynamic state sector (1,2,...,7) defines the present knowledge. The future knowledge is whether the reassignment motion requires a motion further forward in the same direction, whether stopping short is possible, or whether a reversal in direction will be required.

In order to concatenate the nominal trajectory with the reassignment trajectory, the finite jerk limit must be observed, therefore the accelerations must match at the end of all sectors. It is given without proof that the only inputs that are allowed are those of maximum jerk or none at all. The known maximum acceleration limit and velocity limit must also be observed. The normal way of using this information with the required new target information would be the nonlinear solution to a two-point boundary problem with inequality constraints. The problem is nonlinear because the total time to perform the new move is unknown, therefore, time is not an independent variable. Thus the solution to the two-point boundary problem will be chosen to be the reverse of that normally performed, i.e., a

solution will be assumed and the similarity to other known optimal moves will be used as proof.

The construction of reassignment trajectories will be by the construction of a family of trajectories that match up with the nominal trajectory condition of velocity, acceleration, and jerk. The family of curves may be unique in the manner in which a solution is realized. For example, a trivial move as described in the last section may require only a incremental increase in the duration of constant velocity which would be effected by delaying the application of the case 3, sector 5 negative jerk input. A similar technique of delaying the application of jerk, with dramatically different results, will be presented in the next chapter for a non-trivial reassignment trajectory.

Since this author has not found a closed form solution to most of these two-point boundary problems the method of finding a solution will be that of a combination of incrementing a delay of application of an input by some time kT, or cutting short the duration of a dwell. Thus the solutions will be done recursively with increasing kT and the reassignment trajectory will be calculated from the area under the velocity curve. Since the velocity curve is composed of simple shapes (parabolas, trapezoids, rectangles or their combinations), the area or final resulting motion may be easily summed. The real time decision is then to access the proper jerk switch times addressed by the

required relative motion of the nominal and reassignment trajectories, a relative motion that is the area under the reassignment curve.

The generation of reassignment trajectories is not as given in chapter III because, as stated, the nominal trajectories are a function of known total trajectory time. The reassignment trajectory has an unknown position sum and final elapsed time until the trajectory is completely calculated. The incremental approach to the proper area under the velocity curve may be performed on-line, but a simpler solution would be to generate all moves off-line. The quantitative detail of all possible moves would be a function of the addressable memory of the real time controller. The reassignment trajectories of the nominal moves require only eight numbers to be stored for a complete move (seven sector durations, and the offset in target position). If four bytes were assigned to each number then for one megabyte of memory, approximately 30,00 moves could be stored.

A very important consideration for the method of construction is that the method is invariant for all joints or even invariant for different machines. The only information needed to be supplied for the generation of all tables is each specific degree-of-freedom's maximum jerk input, maximum acceleration, and maximum velocity. The next chapter will give an example of how the tables are generated

for the specific actuator constraints and manipulator dynamic parameters. The results of reassignment trajectories being applied to nominal moves will also be shown for an example of a non-trivial trajectory reassignment that requires a reversal in the direction for the degrees-of-freedom.

CHAPTER V
EXAMPLES OF TRAJECTORY REASSIGNMENT

This chapter will offer results of the linearizing and decoupling feedback for a system to demonstrate that the manipulator subsystems can be operated as independent systems. The specific decoupling and linearizing for a two-degree-of-freedom system will be shown first. Since the subsystems will be shown to operate independently, the reassignment of a nominal trajectory will be given to show several different non-trivial moves for a one-degree-of-freedom system. All of the simulation results will be those generated by the models developed in Appendix A and B and with the computer listings in Appendix C. The main programs in Appendix C are for the simulation language Advanced Continuous Simulation Language (ACSL) and the subroutines are written in FORTRAN 77.

Nominal Trajectories

Simulation results for two-degree-of-freedom systems with perfect knowledge of the actuator link parameters and external forces are developed first from the manipulator dynamics. The system modeled is that of a two link pendulum system which exhibits a high degree of coupling and nonlinear properties. Each of the joints is driven by a direct current permanent magnet motor that is armature

controlled. It is assumed that the motor parameter or armature current is directly measurable for each actuator as are the transformed states of acceleration, velocity and position for each joint.

Two tests will be shown to demonstrate the feedback linearization and decoupling transformation on a two-degree-of-freedom system. The first test's nominal motion is that each of the two joints are commanded to move from -1.00 radians to 0.00 radians. The initial and final endpoints are to be at rest, and the specific manipulator dynamic parameters are given in Appendix B. The actual test results are for the following maximum calculated jerk inputs and the state variable constraints:

$$\Delta_1 = \frac{a_{m_1}}{4L_1} * R_1 = 1375 \text{ rad/sec}^3 \tag{5.1}$$

$$\Delta_2 = \frac{a_{m_2}}{4L_2} * R_2 = 6600 \text{ rad/sec}^3 \tag{5.2}$$

where

$$a_{m_1} = 55.0 \text{ rad/sec}^2 \qquad \text{(given)} \tag{5.3}$$

$$a_{m_2} = 55.0 \text{ rad/sec}^2 \qquad \text{(given)} \tag{5.4}$$

$$R_1 = 1.0 \text{ ohms} \tag{5.5}$$

$$R_2 = 0.6 \text{ ohms} \tag{5.6}$$

$$L_1 = 0.01 \text{ henries} \tag{5.7}$$

$$L_2 = 0.005 \text{ henries} \tag{5.8}$$

The maximum angular velocity for each joint is found by

$$\omega_{m_1} = \frac{Vr_1}{Kv_1 \times r_1} = 3.4 \text{ rad/sec} \tag{5.9}$$

$$\omega_{m_2} = \frac{Vr_2}{Kv_2 \times r_2} = 3.4 \text{ rad/sec} \tag{5.10}$$

where

$$Vr_1 = 170.0 \text{ volts} \tag{5.11}$$

$$Vr_2 = 85.0 \text{ volts} \tag{5.12}$$

$$Kv_1 = 0.5 \text{ volts/rad/sec} \tag{5.13}$$

$$Kv_2 = 0.25 \text{ volts/rad/sec} \tag{5.14}$$

$$r_1 = 100 \text{ rad/rad} \tag{5.15}$$

$$r_2 = 100 \text{ rad/rad} \tag{5.16}$$

The figures 5.1-5.5 show the results of the nominal trajectory for each joint from -1.00 radians to 0.00 radians. The input jerks (figure 5.1) are those necessary to force the triple integrator linear system to the final position in minimum time while not exceeding the specified maximum values of acceleration and velocity. The FLDT is successfully transforming the nonlinear coupled manipulator to a linear one as seen by the joint angle profiles in figure 5.2, (θ_1=TH1, θ_2=TH2). The joint angles show a smooth

progression from the initial offset position to the final target as if each joint is operating independently. The angular velocity profiles of the two joints, shown in figure 5.3, demonstrate the characteristic linear triple integrator profile with seven distinct sectors. There are sectors of parabolic increasing (decreasing), linearly increasing (decreasing), and constant velocity. The angular acceleration (figure 5.4), (Y31=α_1, Y32=α_2) also shows the characteristic trapezoidal profile of linear systems with acceleration constraints.

Figure 5.5, however, shows the highly nonlinear input voltage profiles for both actuators during this nominal trajectory. This voltage is the sum of the input jerk and the feedback linearizing and decoupling transformation used to force the two joints to act independently. The resulting armature current (I_1=CUR1, I_2=CUR2) for each of the actuators in shown in figure 5.6. This particular case of nominal trajectories illustrates that the armature closely resembles the joint angular acceleration.

The next and last test for the nominal trajectories requires the first joint to move in minimum time from -1.00 radians to 0.00 radians as before. However, the second joint is commanded to move in minimum time from -0.50 radians to 0.00 radians. Figure 5.8 demonstrates this different motion. The jerk inputs (figure 5.7) for the two joints are now very different in width and the total

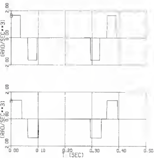


Figure 5.1 Jerk Inputs for TOPC
Joint 1 Trajectory -1.0 to 0.0 Rad
Joint 2 Trajectory -1.0 to 0.0 Rad

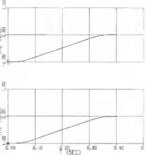


Figure 5-2 Position Trajectory for TOPC
Joint 1 Trajectory -1.0 to 0.0 Rad
Joint 2 Trajectory -1.0 to 0.0 Rad

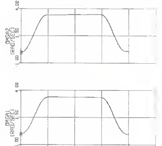


Figure 5.3 Velocity Profile for TOPC
Joint 1 Trajectory -1.0 to 0.0 Rad
Joint 2 Trajectory -1.0 to 0.0 Rad

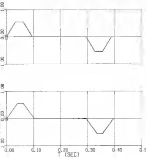


Figure 5.4 Acceleration Profile for TOFC
Joint 1 Trajectory -1.0 to 0.0 Rad
Joint 2 Trajectory -1.0 to 0.0 Rad

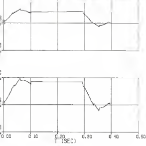


Figure 5.5 Armature Voltage for TOPC
Joint 1 Trajectory -1.0 to 0.0 Rad
Joint 2 Trajectory -1.0 to 0.0 Rad

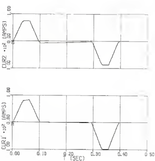


Figure 5.6 Armature Current for TOPC
Joint 1 Trajectory -1.0 to 0.0 Rad
Joint 2 Trajectory -1.0 to 0.0 Rad

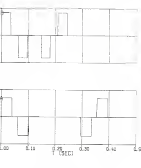


Figure 5.7 Jerk Inputs for TOPC
Joint 1 Trajectory -1.0 to 0.0 Rad
Joint 2 Trajectory -0.5 to 0.0 Rad

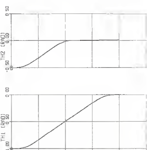


Figure 5-8 Position Trajectory for TOPC
Joint 1 Trajectory -1.0 to 0.0 Rad
Joint 2 Trajectory -0.5 to 0.0 Rad

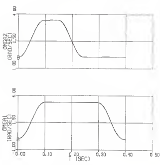


Figure 5-9 Velocity Profile for TOPC
Joint 1 Trajectory -1.0 to 0.0 Rad
Joint 2 Trajectory -0.5 to 0.0 Rad

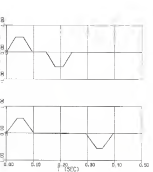


Figure 5.10 Acceleration Profile for TOPC
Joint 1 Trajectory -1.0 to 0.0 Rad
Joint 2 Trajectory -0.5 to 0.0 Rad

duration for each joint to reach its target is also different. The acceleration profile (figure 5.10) of each joint is of similar shape, but of different duration since the two moves require different time to complete. The velocity profiles for the two joints are likewise similar in figure 5.9. However, in figure 5.11 and 5.12 the FLDT voltage and resulting armature current respectively are very different. This difference occurs because the first joint has (for a portion of its trajectory) the added, fixed inertia of the second link.

The FLDT is assumed to be working correctly because the resulting angular position trajectory of each joint is moving as if it were independent. Since the FLDT can now be demonstrated to perform as claimed for a coupled nonlinear two- degree-of-freedom system, all further tests will involve tests with a nonlinear single-degree-of-freedom system without loss of generality

Trajectory Reassignment

Simulation results for a one-degree-of-freedom system (model from Appendix A) with perfect system knowledge and FLDT compensation will be presented to demonstrate that reassignment of a nominal trajectory can be preformed. The system is assumed to have known link parameters and known external forces so that the linearizing feedback is operational. Therefore the optimal controller task is to output minimum time inputs to a triple integrator system

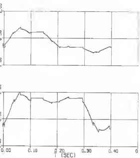


Figure 5.11 Armature Voltage for TOPC
Joint 1 Trajectory -1.0 to 0.0 Rad
Joint 2 Trajectory -0.5 to 0.0 Rad

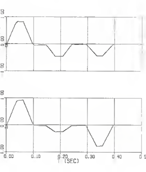


Figure 5.12 Armature Current for TOPC
Joint 1 Trajectory -1.0 to 0.0 Rad
Joint 2 Trajectory -0.5 to 0.0 Rad

with known constraints that should not be exceeded. The states of the triple integrator system are assumed to be position, velocity, and acceleration.

The reassignment trajectories will all be applied to a nominal move of from -1.00 radians to 0.00 radians. The goal of 0.00 radians for this system will be changed during a period of negative acceleration in the nominal move; that is, in sector 6. The nominal move has been made long enough so that a case 3 type move is active where there will be constrained acceleration and constrained velocity during the nominal trajectory. If reassignment occurs during this latter portion of the nominal trajectory, which requires at least stopping short of the original goal, the minimum time reassignment will require a negative velocity (ie. backing up). The new system of input jerks will be required to maintain the maximum rate of negative acceleration for as long as possible in order to reverse the previous direction as soon as possible. The reassignments that will be presented will represent ranges of reassignment for the particular nominal motion where successively more state variable inequality constraints will be reached during the required motion backward. All of the applied jerks during the reassignment will be delivered as if picked from a pre-calculated table whose addresses are those of the proper initial non-zero state variable conditions (case number and

sector number) and the resulting final motion backward from the nominal goal.

The nominal move for the linearized and decoupled system is shown in figure (5.13), the system moves from a static initial position of -1.00 radians to a static finial position of 0.00 radians. The optimal jerk inputs (Z1) were determined as in chapter 3 for TOPC for a system with known constraints on acceleration, velocity and known actuator dynamics. These constraints were calculated by measuring the maximum obtainable acceleration (a_m) of the system. The maximum jerk from the linear model of the actuator armature was calculated by assuming that the armature current will require four time constants to reach maximum value. The maximum jerk was represented by

$$\Lambda = a_m \times \Omega/(4L) = 875 \text{ rad/sec}^3 \tag{5.17}$$

where

Ω = actuator armature resistance (ohms) (5.18)

L = actuator armature inductance (henries) (5.19)

The maximum velocity of the joint was then calculated by

$$\omega_m = V_r/(K_v \times r) \tag{5.20}$$

where

V_r = maximum rated armature voltage (volts) (5.21)

K_v = actuator armature back emf constant [illegible]
(volts/rad/sec)

r = gear box reduction ratio, w_{in}/w_{out} [illegible]
(rad/rad)

The height of the input jerk is a function of the time required to reach maximum acceleration; this corresponds to the fact that the rate of change of current cannot be infinite to do so would require infinite voltage. The input jerk is calculated in an open loop manner as in chapter III and is applied depending, not upon the actual state of the acceleration but on an elapsed time basis.

The nominal trajectory for this one-degree-of-freedom shows the same characteristic state profiles as in the two-degree-of-freedom cases. The position (figure 5.13) of the link (θ-TH1) exhibits a smooth motion from -1.00 radians to 0.00 radians in a minimum time manner. The acceleration profile (figure 5.14) shows regions of linearly increasing, decreasing, and constant magnitude consistent with a case 3 trajectory. The velocity profile (figure 5.14) reflects the regions of constant acceleration (where no jerk is applied) by increasing or decreasing with a constant slope of magnitude a_s. All reassignments for this chapter are assumed to occur before the end of the sector 6 in the velocity profile in the nominal trajectory.

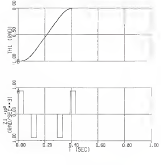


Figure 5.13 Jerk Inputs and Position Trajectory for MSTR
Nominal Trajectory 1-0 to 0-0 Rad

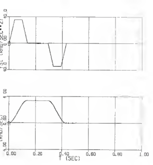


Figure 5.14 Velocity and Acceleration Profile for MSTR Nominal Trajectory -1.0 to 0.0 Rad

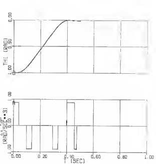


Figure 5.15 Jerk Inputs and Position Trajectory for MSTE Trajectory Reassignment -0.01

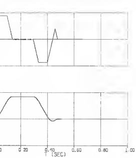


Figure 5.16 Velocity and Acceleration Profile for MSTR Reassignment Trajectory -0.01 Rad

The first reassignment example (figures 5.15 and 5.16) is the case where the reassignment occurs at the end of the linearly decreasing velocity sector, sector 6. The system is committed to apply full negative acceleration to the system and delay the reduction (in magnitude) of this acceleration so that the system crosses over the zero velocity line with some negative acceleration in order to maintain the motion backward. A similar example is the motion of a piston in a reciprocating engine which experiences its maximum acceleration at the top-dead-center of its travel while simultaneously having zero velocity.

The required input jerks were precalculated in an open loop manner based upon the construction of a family of curves with the same initial maximum negative acceleration and then concatenated with the nominal trajectory at the end of the linear negative rate of change of velocity sector, sector 6 (see figure 5.15). Since a delay of the reduction in the maximum negative acceleration will result in the reversal in the system, a table of resulting negative position trajectories can be generated by an increasing time delay. Each of the positive and negative trajectories could contain the same type of required input jerk durations in order to realize a motion that adhered to the state variable inequality constraints while satisfying a particular position motion.

The class of resulting backward moves ranging from moves resulting in unconstrained motion, to that of constrained acceleration and constrained velocity, can be determined by checking whether or not the delay results in the system velocity trajectory reaching milestones ("cases" as in nominal moves). These milestones are a motion produced by a delay in the application of a sector 7 jerk which requires:

1) less than maximum acceleration at the zero velocity crossover point

2) enough delay so that maximum acceleration will be required at or beyond the zero velocity crossover point

3) enough delay so that up to a maximum velocity will be reached

4) enough delay so that a portion of constant negative velocity will be required in the middle of the backward motion.

It must be remembered that all of these cases can be determined by an analysis of the effect of increasing the delay on the velocity profile with the apriori known values of maximum acceleration, velocity, and jerk. The summing of the areas under the velocity profile and the stepping through the constraints will be shown for a case that requires a complete reversal of the nominal trajectory. The software to generate the table of jerk application times can be found in the appendix.

In figures 5.15 and 5.16 the case is illustrated where the reassignment is so small (from the original target) as

to not require the maximum negative acceleration to be applied at the zero velocity point. That is, the maximum rate of negative acceleration will be reduced prior to the crossover from positive to negative acceleration. As one can see in figure 5.15 the applied jerks to the system for this -0.010 reassignment are very different than those of a nominal trajectory, as the second positive jerk is much wider (longer in duration) than the previously applied first positive jerk. In the nominal trajectory these jerks were identical in duration. Additionally, there is another negatively applied jerk appearing in figure 5.15 which was required to slow down and stop the system while it was traversing with negative velocity, (figure 5.16). However, this reassignment does exhibit a negative velocity portion of the trajectory, characteristic of a reassignment during a nominal trajectory, and a positive spike of acceleration to slow the system down and stop it while traveling backwards. The first milestone is demonstrated in this move by the maximum acceleration being reduced by a jerk input before the zero velocity crossover point has been reached.

Figures (5.17) and (5.18) show a target reassignment of -0.10 radians less than the original target of 0.00 radians. The reassignment is large enough (in magnitude) to require the continued maximum negative acceleration (figure 5.18) to be applied to the system as the zero velocity crossover point is reached. This maximum negative

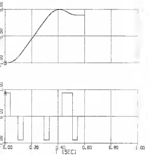


Figure 5.17 Jerk Inputs and Position Trajectory for MSTR Trajectory Reassignment -0.10

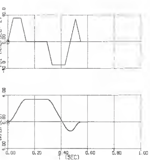


Figure 5.15 Velocity and Acceleration Profile for MSTM Reassignment Trajectory -0.10 Rad

acceleration continues for a short time after this crossover point and requires a corresponding area of positive acceleration to slow the system down and stop it with a final position of -0.15 radians. The reassignment target is far enough away from the nominal target that the position trajectory is noticeably reduced from 0.00 radians after 0.40 seconds during the reassignment portion of the trajectory.

The next reassignment (figures 5.19 and 5.20) requires a final target of -0.35 radians from the nominal target. The required motion backwards is of such magnitude (figure 5.19) that the input jerks drive the system to a state of constant acceleration during the motion backwards. There is a sector of no applied jerk and constant maximum acceleration (figure 5.20) in the reversal similar to that in the nominal portion of the trajectory. The reassignment milestone of requiring maximum acceleration but no constant velocity applies to this type of reassignment motion. The next reassignment will sever all those reassignment motions that require constant velocity portions which form the majority of motions for manipulators.

Figures 5.21 and 5.22 demonstrate the final case where both the inequality constraints on acceleration and velocity are reached. In particular a complete reversal has been commanded back to -1.00 radians after receiving the new assignment in sector 6 of the nominal trajectory. There are

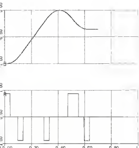


Figure 5.19 Jerk Inputs and Position Trajectory for MSTR Trajectory Reassignment -0.35

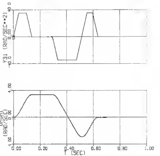


Figure 5.20 Velocity and Acceleration Profile for MSTR Reassignment Trajectory -0.35 Rad

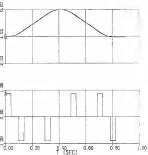


Figure 5.21 Jerk Inputs and Position Trajectory for MSTR
Trajectory Reassignment -1.0
Complete Reversal

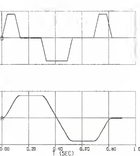


Figure 5.22 Velocity and Acceleration Profile for MSTR Reassignment Trajectory -1.0 Rad Complete Reversal

periods in figure 5.22 of constant rate of change of velocity (constant acceleration), and a period of constant velocity (zero acceleration), during all of these periods there is no jerk applied to the system. This reassignment trajectory has been chosen to analyze for construction of the delay of the sector 7 jerk and the corresponding reversal of the system. The velocity profile for this complete reversal of the manipulator link shows a high degree of symmetry as if the reassignment came first and the nominal trajectory came last.

The original nominal velocity trajectory of the motion from -1.00 radians to 0.00 radians is superimposed in figure 5.23 upon the reassignment trajectory velocity of from -1.00 radians and back to -1.00 radians. The delay of the sector 7 jerk can only be delayed up to the point where the maximum negative acceleration must be reduced to avoid exceeding the maximum allowable negative velocity. The complete reversal of this nominal trajectory could not be satisfied by an additional dwell of zero acceleration and constant negative velocity. The construction of the tables of reassignment relative position motions and corresponding jerk application time automatically accounts for this fact. The reassignment velocity trajectory is constructed (as shown in this figure 5.23) of several easily analyzed areas. Note that the sum of the areas, both positive and negative, result in the relative motion of the reassignment trajectory.

The areas under the velocity profile are a function of the known inequality constraints of the manipulator for this particular degree-of-freedom. The slopes of the constant acceleration portion of the linearly increasing/ decreasing (in magnitude) velocity profile are a_m. The time to increase/ decrease the acceleration to its maximum is a function of a_m and Δ (as in the nominal trajectory) is

$$t_1 = \frac{a_m}{\Delta} \qquad (5.24)$$

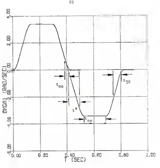
The corresponding time to reach the crossover point (before which the system is still approaching the old target) and time durations of the different sectors follows. Note that the times are for a sector duration (and consequent jerk times) not as shown in chapter iii tables for solution of the TOPC control problem.

Corresponding to figure 5.23 the following times can be calculated from known incremental values of delay in sector 7 jerk and additional dwell time, this total increment is in units of kT (sec).

$$\omega_0 = \frac{\Delta t_1^2}{2} \qquad (5.25)$$

$$t_{oo} = \frac{\omega_0}{a_m} \qquad (5.26)$$

$$\omega^* = \omega_m - \omega_0 \qquad (5.27)$$



$$\omega_0 = \frac{At_1^2}{2} = \frac{\omega_m^2}{2A} \qquad t_{co} = \frac{\omega_0}{a_m}$$

$$\omega^* = \omega_m - \omega_0 \qquad t^* = \frac{\omega^*}{a_m}$$

$$t_{cv} = kT - t^* - t_{co} \qquad t_{32} = \frac{T(\omega_m - 2\omega_0)}{\bar{a}_m}$$

Figure 5.23 Superposition of Nominal and Reassignment Velocity Profile with Reassignment Sectors

$$t^* = \frac{v^*}{a_m} \qquad (5.28)$$

$$t_{ov} = kT - t^* - t_{co} \qquad (5.29)$$

$$t_{32} = \frac{1}{a_m}(v_m - 2a_0) \qquad (5.30)$$

The position resulting from this kT unit delay is a linear sum of all areas under the velocity profile generated by the known velocities and time intervals. The jerks required to effect this profile are then the associated times necessary to reach maximum acceleration, maximum velocity, and dwell at constant velocity. The jerks which reduce the velocity back to zero are those identical to the solution in chapter IV, although they were generated in a very different manner. With each increase in kT during the construction stage, a new set of jerks and total area would be calculated. Each is referenced by the match to the case (1, 2, or 3) and sector (1,2,...,7) in which the reassignment occurred (note that this uniquely determines the concatenated states of velocity and acceleration), and the required relative motion of the reassignment to the nominal target.

When a reassignment is encountered the general family of moves can be chosen as above, however, a method of concatenating the new jerks with those applied must be done in a manner consistent with the observation that all moves have seven sectors. Since elapsed relative time is the only

controlling variable in the real time decision controller (with respect to when to apply a positive, negative, or no jerk input to each actuator) the reassignment jerk dwell times must be reoriented to the seven sector pattern. In the specific case of a reassignment requiring a motion backwards after the reassignment demand has come in sector 6, the just calculated sixth (6^{th}) sector becomes the new first (1^{st}) sector. The other times are assigned accordingly in reference to figure 5.23

$$\hat{t}_1 = t_6 + kT_m \tag{5.31}$$

$$\hat{t}_2 = 0.0 \tag{5.32}$$

$$\hat{t}_3 = t_1 \tag{5.33}$$

$$\hat{t}_4 = t_{ov} \tag{5.34}$$

$$\hat{t}_5 = t_1 \tag{5.35}$$

$$\hat{t}_6 = t_{32} \tag{5.36}$$

$$\hat{t}_7 = t_1 \tag{5.37}$$

Note that the reassignment times have formed the new nominal trajectory switch times in which additional reassignments may occur.

When using this method of generating (and addressing) the required jerks for a specific reassignment, the intelligent multi-actuator controller can be realized. The optimal

control solution is a combination of off line generation (which could be carried out at start up time due to its efficiency) and real time decision making that applies positive, negative or no voltages which represent jerk inputs. The controller uses existing conditions to determine which optimal input to apply to triple integrator systems. All of the inputs are constructed with the a priori knowledge of actuator parameters and system dynamics. A summary of the results of this dissertation will follow in the next chapter.

CHAPTER VI
CONCLUSIONS AND FUTURE RESEARCH

This dissertation demonstrates the only known solution to the minimum time control with midstream trajectory reassignment of a multi degree-of-freedom system. A general model for N-degree-of-freedom systems was shown that included the actuator dynamics. A feedback linearizing and decoupling transformation (FLDT) was used, from previous work, to force each degree-of-freedom to be a linear triple integrator system.

The time optimal positioning control (TOPC) for static endpoints, also from previous work, was used as a basis for analyzing how to approach the construction and concatenation of reassignment trajectories. The problem of reassignment was then analyzed by the segmentation of all nominal trajectories into general cases of active state variable constraints. The reassignment possibilities were further analyzed as to when a reassignment could occur within a nominal trajectory. Then each nominal trajectory was broken down into seven velocity sectors. Based upon when the reassignment ocurred in the seven sectors, the reassignment was shown to require different solutions. The solutions to reassignment could be one of continued forward motion, or stopping short of the current nominal trajectory, or

the reversal of the forward motion. The reassignment switch times then become a nominal trajectory's switch times in which additional reassignments could occur.

The decision of what reassignment input jerks to apply to the triple integrator system was shown to be the result of an off line solution to incremental additions in areas under simple velocity profiles. These velocity profiles were produced from the known actuator performance parameters and manipulator dynamics. These parameters were identical to those used in the TOPC for the generation of nominal trajectory solutions with static endpoints.

The structure of the solutions was intended for real time implementation in an intelligent controller faced with alternative decisions for minimum time trajectory planning. The construction of jerk inputs for reassignment trajectories was developed as a universal method based upon dynamic parameters of maximum acceleration and actuator motor constants. Because of the universal method of determining minimum time solutions with trajectory reassignment, the methods are applicable to machines not thought of as manipulators, such as machine tools and autonomous vehicles.

Future research that would extend this work would complete the definition of all possible reassignment trajectories with multiple link systems. The generation of jerk switch tables and real time decision making software

should be implemented in a real system for performance evaluation and simulation validation. The work should be extended to those system that are fourth order in each of the degrees of freedom which would require trapezoidal jerk inputs. The extension to fourth order systems would then include solutions to systems that had significant dynamics in the current amplifiers for each actuator. The fourth order solutions could also be extended to those manipulator systems that did not have ridgid links but actuator dynamics as described in this work.

The most promising area for future research would be the use of trajectory reassignments for obstacle avoidance during minimum time motion. The optimal solution of a minimum time problem of third order (or higher) with state variable constraints including that on the position state variable are not known. The manner in which the reassignment trajectories for this dissertation have been developed should prove useful in the avoidance problem. Real world application of this avoidance problem coupled, with the minimum time trajectory reassignment, would be in autonomous mobile systems.

APPENDIX A
ONE-DEGREE-OF-FREEDOM SYSTEM MODEL AND FLDT

The one degree-of-freedom system is an rigid body pendulum rotating about a fixed axis, with point mass at the end of the link, driven by a permanent magnet direct current motor. The system model is developed from Lagrangian mechanics for rigid bodies from the expressions for kinetic and potential energies of the system. The kinetic energy T, of the system is

$$T = \frac{1}{2} m \times d^2 \times \omega^2 \tag{A.1}$$

and the potential energy V, of the system is found by

$$V = -m \times g \times d \times \cos(\theta) \tag{A.2}$$

where

- m   mass of the link, centered at the center of gravity (kg)
- g   acceleration due to gravity (m/sec^2)
- d   center of gravity of link from origin, assumed at end of link (m)
- θ generalized coordinate of link, measured from perpendicular (rad)

$\dot{\theta}$ first derivative of generalized coordinate, velocity of the link (rad/sec).

The Lagrangian function L, for non-conservative systems is written as

$$L = T - V \tag{A.3}$$

The generalized force F_θ for the Lagrangian system is formed from the Lagrangian by

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} - F_\theta \tag{A.4}$$

The necessary differentiation to form the Lagrangian equation (A.4) is

$$\frac{\partial L}{\partial \theta} = \frac{\partial T}{\partial \theta} - \frac{\partial V}{\partial \theta} \tag{A.5}$$

where

$$\frac{\partial T}{\partial \theta} = 0 \tag{A.6}$$

$$\frac{\partial V}{\partial \theta} = m*g*d*\sin(\theta) \tag{A.7}$$

Then

$$\frac{\partial L}{\partial \theta} = -m*g*d*\sin(\theta) \tag{A.8}$$

The partial with respect to the velocity is

$$\frac{\partial L}{\partial \omega} = \frac{\partial T}{\partial \omega} - \frac{\partial V}{\partial \omega} \qquad (A.9)$$

where

$$\frac{\partial T}{\partial \omega} = m*d^2*\omega \qquad (A.10)$$

$$\frac{\partial V}{\partial \omega} = 0 \qquad (A.11)$$

Then

$$\frac{\partial L}{\partial \omega} = m*d^2*\omega \qquad (A.12)$$

Differentiating with respect to time

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \omega} \right) = m*d^2*a \qquad (A.13)$$

where

a acceleration, the derivative of the velocity (rad/sec^2)

The generalized force may be formed by the difference of equations (A.13) and (A.8) as

$$F_\theta = m*d^2*a + m*g*d*\sin(\theta) \qquad (A.14)$$

The differential equation for the actuator follows the convention of chapter II, equation (2.2). The generalized force F_θ is the generalized loading torque $\underline{T}$ in equation

(A.7), so the sum of the torques in the joint may be written as

$$J_m r^2 \alpha + F_B = K_T r I - A\theta - B\omega \qquad (A.15)$$

where

$A = 0$ compliance coefficient

$B = 0$ viscous friction

Substituting the generalized force equation (A.14) into (A.15) and solving for the joint acceleration α

$$[J_m r^2 + md^2]\alpha = K_T r I - mgd\sin(\theta) \qquad (A.15)$$

$$\alpha = \frac{1}{[J_m r^2 + md^2]}\{ K_T r I - mgd\sin(\theta) \} \qquad (A.16)$$

Similar to equation (2.3), the rate of change of current in the motor may be written as

$$I' = -\frac{RI}{L} - \frac{K_b r\omega}{L} + \frac{V}{L} \qquad (A.17)$$

Using the state definitions in chapter II for the link the other state equations along with (A.17) may be formed as

$$\theta' = \omega \qquad (A.18)$$

$$\dot{\omega} = \frac{1}{[J_m r^2 + md^2]} \{ K_T r I - mgd\sin(\theta) \} \tag{A.19}$$

The linearizing and decoupling feedback transformation voltage V (equation 2.25) may be formed by using the state identities (2.4) and (2.5) and substitution of the appropriate equations from (A.17-A.19) as

$$J^*(\underline{Y}_1,t) = [J_m r^2 + md^2] \tag{A.20}$$

$$G^* = 0 \tag{A.21}$$

$$l(\underline{Y}_1,\underline{Y}_2,t) = mgd\sin(\theta) \tag{A.22}$$

$$\frac{d}{dt}[\, l(\underline{Y}_1,\underline{Y}_2,t) \,] = mgd\cos(\theta)\omega \tag{A.23}$$

$$\underline{V}(\underline{Y},\underline{Z},t) = L\bar{K}_T^{-1}[\, J^*(\underline{Y}_1,t)\underline{Z} + (\, \frac{d}{dt}[J^*(\underline{Y}_1,t)] + \bar{R}J^*(\underline{Y}_1,t)\,)\underline{Y}_3$$

$$+\frac{d}{dt}[\, l(\underline{Y}_1,\underline{Y}_2,t) \,] + (\, \bar{K}_T\bar{K}_V \,)\underline{Y}_2 + \bar{R}(\, \underline{l}(\underline{Y}_1,\underline{Y}_2,t) \,) \,\} \tag{2.24}$$

This voltage was applied to the one-degree-of-freedom system in chapter V for the reassignment examples. The input jerks Z = $\underline{Z}$ were determined by the proper reassignment trajectory in the real time decision controller.

APPENDIX E
TWO-DEGREE-OF-FREEDOM SYSTEM MODEL AND PLOT

The two-degree-of-freedom system is an rigid two link pendulum rotating about a fixed axis for joint one, with point masses at the end of the links, driven by two permanent magnet direct current motors. The system model is developed from Lagrangian mechanics for rigid bodies from the expressions for kinetic and potential energies of the system as in appendix A. The kinetic energy T_1, of the first link is

$$T_1 = \frac{1}{2} m_1 * d_1^2 * \omega_1^2 \qquad (E.1)$$

and the potential energy V_1, of the first link is found by

$$V_1 = -m_1 * g_1 * d_1 * \cos(\theta_1) \qquad (E.2)$$

where

- m_1 mass of the link, at the center of gravity (kg)
- g acceleration due to gravity (m/sec^2)
- d_1 center of gravity of link from origin, assumed at end of link (m)
- θ_1 generalized coordinate of link, measured from perpendicular (rad)

ω_1 first derivative of generalized coordinate, velocity of the link (rad/sec).

The second link is analyzed by first expressing the position of the link in Cartesian coordinates as

$$x_2 = d_1\sin(\theta_1) + d_2\sin(\theta_1+\theta_2) \tag{B.3}$$

$$y_2 = -d_1\cos(\theta_1) - d_2\cos(\theta_1+\theta_2) \tag{B.4}$$

Upon differentiating the positions with respect to time in order to derive the velocity of the second link in Cartesian coordinates, the velocities are

$$\dot{x}_2 = d_1\cos(\theta_1)\omega_1 + d_2\cos(\theta_1+\theta_2)(\omega_1+\omega_2) \tag{B.5}$$

$$\dot{y}_2 = d_1\sin(\theta_1)\omega_1 + d_2\sin(\theta_1+\theta_2)(\omega_1+\omega_2) \tag{B.6}$$

Forming the kinetic and potential energy of the second link as the first link was derived

$$T_2 = \frac{1}{2}m_2(x'^2_2 + y'^2_2) \tag{B.7}$$

$$T_2 = \frac{1}{2}m_2d_1^2[\ \omega_1^2 + (\omega_1+\omega_2)^2\ ] + m_2d_1d_2\cos(\theta_2)\omega_1(\omega_1+\omega_2) \tag{B.8}$$

$$V_2 = m_2gy_2 \tag{B.9}$$

$$V_2 = -m_2gd_1\cos(\theta_1) - m_2gd_2\cos(\theta_1+\theta_2) \tag{B.10}$$

where

m_2 mass of the link, centered at the center of gravity (kg)

d_2 center of gravity of link from link 1, assumed at end of link 2 (m)

θ_2 generalized coordinate of link, measured from link 1 (rad)

ω_2 first derivative of generalized coordinate, velocity of the link (rad/sec)

The Lagrangian function L, for non-conservative systems is written as in appendix A as

$$L = T_1 + T_2 - V_1 - V_2 \tag{E.11}$$

The generalized force F_{θ_1} for the first link is formed from the Lagrangian by

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \omega_1}\right) - \frac{\partial L}{\partial \theta_1} = F_{\theta_1} \tag{E.12}$$

The necessary differentiation to form the Lagrangian equation (E.12) is

$$\frac{\partial L}{\partial \theta_1} = \frac{\partial T}{\partial \theta_1} - \frac{\partial V}{\partial \theta_1} \tag{E.13}$$

where

$$T = T_1 + T_2 \tag{E.14}$$

$$V = V_1 + V_2 \tag{E.15}$$

$$\frac{\partial L}{\partial \theta_1} = -(m_1 + m_2)gd_1\sin(\theta_1) - m_2gd_2\sin(\theta_1+\theta_2) \quad (B.16)$$

The partial with respect to the velocity is

$$\frac{\partial L}{\partial \omega_1} = \frac{\partial T}{\partial \omega_1} - \frac{\partial V}{\partial \omega_1} \quad (B.17)$$

$$\frac{\partial L}{\partial \omega_1} = (m_1 + m_2)d_1^2\omega_1^2 + m_2d_2^2(\omega_1+\omega_2)$$
$$+ m_2d_1d_2\cos(\theta_2)(\omega_1+\omega_2) + m_2d_1d_2\cos(\theta_2)\omega_1 \quad (B.18)$$

Differentiating with respect to time

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \omega_1}\right) = [(m_1 + m_2)d_1^2 + m_2d_2^2 + 2m_2d_1d_2\cos(\theta_2)]\alpha_1$$
$$+ [m_2d_2^2 + m_2d_1d_2\cos(\theta_2)]\alpha_2$$
$$-2m_2d_1d_2\sin(\theta_2)(\omega_1\omega_2) - m_2d_1d_2\sin(\theta_2)\omega_2^2$$
$$(B.19)$$

where

α_1 acceleration, the derivative of the velocity of the first joint (rad/sec^2)

α_2 acceleration, the derivative of the velocity of the second joint (rad/sec^2)

The generalized force for the first joint may be formed by the difference of equations (B.18) and (B.19) as

$$F_{\theta 1} = [(m_1 + m_2)d_1^2 + m_2 d_2^2 + 2m_2 d_1 d_2 \cos(\theta_2)]a_1$$

$$+ [m_2 d_2^2 + m_2 d_1 d_2 \cos(\theta_2)]a_2 \quad - 2m_2 d_1 d_2 \sin(\theta_2)(\omega_1 \omega_2)$$

$$- m_2 d_1 d_2 \sin(\theta_2)\omega_2^2$$

$$+ (m_1 + m_2) g d_1 \sin(\theta_1) + m_2 g d_2 \sin(\theta_1 + \theta_2) \qquad (B.20)$$

Similarly for the second link the generalized force $F_{\theta 2}$ may be found as

$$\frac{\partial L}{\partial \theta_2} = -m_2 d_1 d_2 \sin(\theta_2)\omega_1^2 - m_2 d_1 d_2 \sin(\theta_2)\omega_1 \omega_2$$

$$- m_2 g d_2 \sin(\theta_1 + \theta_2) \qquad (B.21)$$

and

$$\frac{\partial L}{\partial \omega_2} = m_2 d_2^2(\omega_1 + \omega_2) + m_2 d_1 d_2 \cos(\theta_2)\omega_1 \qquad (B.22)$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \omega_2} \right) = m_2 d_2^2(a_1 + a_2) + m_2 d_1 d_2 \cos(\theta_2) a_1$$

$$- m_2 d_1 d_2 \sin(\theta_2)\omega_1 \omega_2 \qquad (B.23)$$

The generalized force $F_{\theta 2}$ is then

$$F_{\theta 2} = [\ m_2 d_2^2 + m_2 d_1 d_2 \cos(\theta_2)\ ]\alpha_1 + [m_2 d_2^2\ ]\alpha_2$$

$$+ m_2 d_1 d_2 \sin(\theta_2)\omega_1^2 + m_2 g d_2 \sin(\theta_1 + \theta_2) \qquad (B.24)$$

The differential equations for the actuators follow the convention of chapter II, equation (2.2). The generalized forces $F_{\theta 1}$ and $F_{\theta 2}$ are the generalized loading torques $\underline{T}$ in equation (2.2), so the sum of the torques in the joints may be written with the following definitions

$$J^* = \begin{vmatrix} J_{11} & J_{12} \\ J_{12} & J_{22} \end{vmatrix} \qquad (B.25)$$

where

$$J_{11} = Jm_1 r_1^2 + (m_1 + m_2) d_1^2 + m_2 d_2^2 + 2 m_2 d_1 d_2 \cos(\theta_2) \qquad (B.26)$$

$$J_{12} = m_2 d_2^2 + m_2 d_1 d_2 \cos(\theta_2) \qquad (B.27)$$

$$J_{22} = m_2 d_2^2 + Jm_2 r_2^2 \qquad (B.28)$$

The torque constant matrix is

$$\bar{K}_T = \begin{vmatrix} K_{T1} & 0 \\ 0 & K_{T2} \end{vmatrix} \tag{B.29}$$

The external loading vector is

$$l(\underline{Y}_1, \underline{Y}_2) = \begin{vmatrix} (m_1 + m_2) g d_1 \sin(\theta_1) + m_2 g d_2 \sin(\theta_1 + \theta_2) \\ m_2 g d_2 \sin(\theta_1 + \theta_2) \end{vmatrix} \tag{B.30}$$

The centripetal and coriolis torque weighting vector is

$$G^* = \begin{vmatrix} -2 m_2 d_1 d_2 \sin(\theta_2)(\omega_1 \omega_2) - m_2 d_1 d_2 \sin(\theta_2) \omega_2^2 \\ m_2 d_1 d_2 \sin(\theta_2) \omega_1^2 \end{vmatrix} \tag{B.31}$$

The two additional motor constant matrices are

$$\bar{R} = \begin{vmatrix} \frac{R_1}{L_1} & 0 \\ 0 & \frac{R_2}{L_2} \end{vmatrix} \tag{B.32}$$

$$K_v = \begin{vmatrix} \frac{K_{v1} r_1}{L_1} & 0 \\ 0 & \frac{K_{v2} r_2}{L_2} \end{vmatrix} \qquad (8.33)$$

The following compliance and viscous matrices are assumed

A = [0]   compliance coefficient

B = [0]   viscous friction

The state equation for the position of the links is

$$\underline{\theta}' = \underline{\omega} \qquad (8.34)$$

The state equation for the current in the actuators is

$$\underline{I}' = -\frac{\underline{R}\underline{I}}{\underline{L}} - \frac{\underline{K_v r \omega}}{\underline{L}} + \frac{\underline{V}}{\underline{L}} \qquad (8.35)$$

Using the state definitions in chapter II for the links, the other state equations along with (8.34) may be formed as for the first link in appendix A. The linearizing and decoupling feedback transformation voltage $\underline{V}$ (equation 2.25) may be formed by using the state identities (2.4) and (2.8). Then with substitution of the appropriate equations from above,

$$\underline{V}(\underline{Y},\underline{Z},t) = L\bar{K}_T^{-1}\{\, J^*(\underline{Y}_1,t)\underline{Z} + \left(\frac{d}{dt}[J^*(\underline{Y}_1,t)\ \text{[illegible]} \right.$$

$$\left. +\frac{d}{dt}[\ 1(\underline{Y}_1,\underline{Y}_2,t)\ ] + (\ \bar{K}_T\bar{K}_V\ )\underline{Y}_2 + \bar{R}(\ \underline{1}(\underline{Y}_1,\underline{Y}_2,t)\ )\ \right\}$$

(E.36)

The following derivatives are found in (E.36)

$$\frac{dG^*}{dt} = \begin{vmatrix} -m_2d_1d_2\{\cos(\theta_2)\omega_2(2\omega_1\omega_2+\omega_2^2)+2\sin(\theta_2)(\alpha_1\omega_2+\omega_1\alpha_2+\omega_2\alpha_2)\} \\ \\ m_2d_1d_2(\ \cos(\theta_2)\omega_2\omega_1^2 + 2\sin(\theta_2)\omega_1\alpha_1\ ) \end{vmatrix}$$

(E.37)

$$\frac{d}{dt}1(\underline{Y}_1,\underline{Y}_2) = \begin{vmatrix} (m_1+m_2)gd_1\cos(\theta_1)\omega_1 + m_2gd_2\cos(\theta_1+\theta_2)(\omega_1+\omega_2) \\ \\ m_2gd_2\cos(\theta_1+\theta_2)(\omega_1+\omega_2) \end{vmatrix}$$

(E.38)

and

$$\frac{d}{dt}[J^*(\underline{Y}_1,t)] = \begin{vmatrix} -2m_2d_1d_2\sin(\theta_2)\omega_2 & -m_2d_1d_2\sin(\theta_2)\omega_2 \\ \\ -m_2d_1d_2\sin(\theta_2)\omega_2 & 0 \end{vmatrix}$$

(E.39)

This voltage $\underline{V}$ was applied to the two-degree-of-freedom system in chapter V for the nominal trajectory examples. The input jerks $\underline{Z}$ were determined by the methods of chapter III for TOPC.

APPENDIX C
COMPUTER PROGRAM LISTINGS

The following computer listings are for the simulations of both one and two-degree-of-freedom systems as modeled in appendix A and B. The main programs are those translated by the simulation language Advanced Continuous Simulation Language (ACSL). These main programs are in a particular format that is unique to ACSL and cannot be compiled by a FORTRAN compiler. The ACSL translator transforms the user's compact programs into efficient FORTRAN which can be linked with other FORTRAN programs and the ACSL libraries. ACSL is available from Mitchell and Gauthier Associates, Inc., 290 Baker Avenue, Concord, Ma 01742. Note that the form of the ACSL listings is

```
INITIAL
    .
    . CONSTANTS
      INITIAL CONDITIONS
    .
END
DYNAMIC
DERIVATIVE
    .
    . DIFFERENTIAL EQUATIONS
END
    TERMINAL CONDITIONS
END
END
```

The other listings are FORTRAN 77 subroutines that are written to determine the jerk switch times for both the nominal and reassignment trajectories.

Advanced Continuous Simulation Language Listing for
Two-Degree-of-Freedom System

```
PROGRAM 2DOF PLANAR MAN. SYS. WITH PLDT & REASSIGNMENT
"TDOF1T.CSL"
INITIAL
  CINTERVAL CINT = 0.005
  MAXTERVAL MAXT = 0.00025
  NSTEPS    NSTP = 1
  CONSTANT  D1   = 1.0 , D2   = 1.0
  CONSTANT  M1   = 10.0, M2   = 10.0
  CONSTANT  G    = 9.8
  CONSTANT  R1   = 100.0, R2   = 100.0
  CONSTANT  RES1 = 1.0 , RES2 = 0.6
  CONSTANT  TND1   = 0.00025 , TND2   = 0.00015
  CONSTANT  A11   = 0.0 , A22   = 0.0
  CONSTANT  E11   = 0.0 , E22   = 0.0
  CONSTANT  KT1  = 0.5 , KT2  = 0.25
  CONSTANT  KV1  = 0.5 , KV2  = 0.25
  CONSTANT  JM1  = 0.002 , JM2 = 0.001
  CONSTANT TH1IC = -1.0, TH2IC = -1.0
  CONSTANT OMGA1C = 0.0, OMGA2C = 0.0
  CONSTANT TSTP  = 1.0
  CONSTANT V1 = 0.0, V2 = 0.0
  CONSTANT Z1 = 0.0, Z2 = 0.0
  CONSTANT MUL1 = 1.0, MUL2 = 1.0
  CONSTANT VR1 = 170.0, VR2 = 85.0
  CONSTANT CUR1M = 100.0, CUR2M = 100.0
  CONSTANT OM1M = 340.0, OM2M = 340.0
  CONSTANT ALPH1M = 55.0, ALPH2M = 55.0
  CONSTANT TH1F = 0.0, TH2F = 0.0
  CONSTANT CLAMPV = 1.5
  CONSTANT CLAMPC = 1.5
  CONSTANT CLAMPA = 1.0
```

```
  T1F = TSTP
  T2F = TSTP
  OMGA1M = VM1 / ( KV1 * R1 )
  OMGA2M = VM2 / ( KV2 * R2 )
  DELT1 = ALPH1M* RES1/ ( 4.0* IND1 )
  DELT2 = ALPH2M* RES2/ ( 4.0* IND2 )
  ZETA1 = TH1F - TH1IC
  ZETA2 = TH2F - TH2IC
  C11 = KT1*R1
  C22 = KT2*R2
  F11 = -R1*KV1/IND1
  F12 = -R2*KV2/IND2
  F21 = -RES1/IND1
  F22 = -RES2/IND2
  LC1 = IND1/(KT1*R1)
  LC2 = IND2/(KT2*R2)
  L1 = (M1 +M2 )*G*D1*SIN(TH1) +M2*G*D2*SIN(TH1+TH2)
  L2 = M2*G*D2*SIN(TH1+TH2)
  CUR1IC = L1 / C11
  CUR2IC = L2 / C22
END $" OF INITIAL "
```

```
DYNAMIC
DERIVATIVE
"  Z2 = PULSE( 0.0 , 1.0 , .1 ) - PULSE( .5 , 1.0 , .1 )"
"  Z2 = .5*STEP(.1) -.5*STEP(.6) + .5*STEP(1.1)
"  Z1 = 20.0* STEP(0.0) *MUL1 "
"  Z2 = 20.0* STEP(0.0) *MUL2 "

  TH1   = INTEG( OMGA1 ,TH1IC )
  TH2   = INTEG( OMGA2 ,TH2IC )

   L1 = (M1 +M2 )*G*D1*SIN(TH1) +M2*G*D2*SIN(TH1+TH2)
   L2 = M2*G*D2*SIN(TH1+TH2)

J11=(M1+ M2)*D1*D1 +M2*D2*D2+2.0*M2*D1*D2*COS(TH2)+JM1*H1*H1
   J12 = M2*D2*D2 +M2*D1*D2*COS(TH2)
   J22 = M2*D2*D2 +JM2*H2*H2
   DETJ = J11*J22 -J12*J12

   WP = -M2* D1* D2* SIN(TH2)
   W1 = ( 2.0*OMGA1*OMGA2 + OMGA2*OMGA2 )* WP
   W2 = ( - OMGA1* OMGA1 )* WP

   G1P = C11*CUR1 -L1 -W1 -A11*TH1 -B11*OMGA1
   G2P = C22*CUR2 -L2 -W2 -A22*TH2 -B22*OMGA2

G1 = BOUND( (-ALPH1M * CLAMPA) ,( ALPH1M * CLAMPA) , G1PP )
   G1PP = ( J22*G1P -J12*G2P )/DETJ
G2 = BOUND( (-ALPH1M * CLAMPA) ,( ALPH1M * CLAMPA) , G2PP )
   G2PP = (-J12*G1P +J11*G2P )/DETJ

   OMGA1 = INTEG( G1 , OMGA1C )
   OMGA2 = INTEG( G2 , OMGA2C )

   CUR1D = ( V1/IND1 + F11* OMGA1 + F21* CUR1 )
   CUR2D = ( V2/IND2 + F12* OMGA2 + F22* CUR2 )

CUR1 = LIMINT( CUR1D, CUR1IC ,-CUR1M*CLAMPC , CUR1M*CLAMPC )
CUR2 = LIMINT( CUR2D, CUR2IC ,-CUR2M*CLAMPC , CUR2M*CLAMPC )
" END OF MANIPULATOR DYNAMICS SECTION "
```

```
" BEGINNING OF THE CONTROLLER SECTION "

   Y31 = G1
   Y32 = G2

   WPD = -M2* D1* D2* OMGA2* COS( TH2 )
   W1D = ( 2.0*OMGA1*OMGA2 + OMGA2*OMGA2 )*WPD...
       + 2.0*( Y31*OMGA2 + OMGA1*Y32 + OMGA2*Y32 )*WP
   W2D = -2.0*OMGA1*Y31*WP -OMGA1*OMGA1*WPD

   L1D = (M1+M2)*G*D1*OMGA1*COS(TH1)...
       + M2*G*D2*(OMGA1 + OMGA2)*COS(TH1+TH2)
   L2D = M2*D*D2*(OMGA1 + OMGA2)*COS(TH1+TH2)

   J12D = -M2*D1*D2*OMGA2*SIN(TH2)
   J11D = 2.0*J12D
   J22D = 0.0

V1 = BOUND( -VR1*CLAMPV , VR1*CLAMPV , V1P )
V1P=LC1*( J11*Z1+J12*Z2 +W1D +L1D -F21*(W1+L1)+...
(-C11*F11)*OMGA1 +(-F21*J11+J11D)*Y31 +(J12D-F21*J12)*Y32 )

V2 = BOUND( -VR2*CLAMPV , VR2*CLAMPV , V2P )
V2P=LC2*( J12*Z1+J22*Z2 +W2D +L2D -F22*(W2+L2)+...
(-C22*F12)*OMGA2 +(J12D-F22*J12)*Y31 +(-F22*J22+J22D)*Y32 )

PROCEDURAL(Z1=ZETA1,DELT1,T,TH1IC,OMGA1M,ALPH1M,T1F)
CALL OPT1(Z1,ZETA1,DELT1,T,TH1IC,OMGA1M,ALPH1M,T1F)
END $" OF PROCEDURAL1 "

PROCEDURAL(Z2=ZETA2,DELT2,T,TH2IC,OMGA2M,ALPH2M,T2F)
CALL OPT2(Z2,ZETA2,DELT2,T,TH2IC,OMGA2M,ALPH2M,T2F)
END $" OF PROCEDURAL2 "

END $" OF DERIVATIVE "
   TERMT(T.GE.TSTP)
END $" OF DYNAMIC "
END $" OF PROGRAM "
```

Switch Time Generator for Joint 1

```
SUBROUTINE OPT1(Z1,ZETA1,DELT1,T,TH1IC,OMGA1M,ALPH1M,T1F)
IMPLICIT NONE
REAL*4 Z1
REAL*4 ZETA1
REAL*4 DELT1
REAL*4 T
REAL*4 TH1IC
REAL*4 OMGA1M
REAL*4 ALPH1M
REAL*4 T1F
REAL*4 T1FF
REAL*4 T1FFF
IF ( ZETA1 .LT. ( 2.0*ALPH1M**3/(DELT1*DELT1) ) ) THEN
     T1F = 2.0*( 4.0*ZETA1/DELT1 )**(1.0/3.0)
   IF ( ( 0.0 .LE. T ) .AND. ( T .LT. (T1F/4.0) ) ) THEN
           Z1 = DELT1
   ELSEIF ( ( (T1F/4.0) .LE. T ) .AND.
&          ( T .LT. (3.0*T1F/4.0) ) ) THEN
           Z1 = -DELT1
   ELSEIF ( ( (3.0*T1F/4.0) .LE. T ) .AND.
&          ( T .LE. T1F ) ) THEN
           Z1 = DELT1
   ELSE
           Z1 = 0.0
   ENDIF
ELSEIF(( ZETA1 .GT. ( 2.0*ALPH1M**3/(DELT1*DELT1) )).AND.
& (OMGA1M.GT.(SQRT(ALPH1M**4/(4.0*DELT1*DELT1)+ALPH1M*ZETA1)
& -ALPH1M**2/(2.0*DELT1) ) ) ) THEN
T1FF=ALPH1M/DELT1+ SQRT((ALPH1M/DELT1)**2+4.0*ZETA1/ALPH1M)
   IF( ( 0.0 .LE. T ) .AND. ( T .LT. (ALPH1M/DELT1) ))THEN
           Z1 = DELT1
   ELSEIF ( ( (ALPH1M/DELT1) .LE. T ) .AND.
&          ( T .LT. (T1FF/2.0-ALPH1M/DELT1 ) ) ) THEN
           Z1 = 0.0
   ELSEIF ( ( (T1FF/2.0-ALPH1M/DELT1 ) .LE. T ) .AND.
&          ( T .LT. (T1FF/2.0+ALPH1M/DELT1 ) ) ) THEN
           Z1 = -DELT1
   ELSEIF ( ( (T1FF/2.0+ALPH1M/DELT1 ) .LE. T ) .AND.
&          ( T .LT. (T1FF -ALPH1M/DELT1 ) ) ) THEN
           Z1 = 0.0
   ELSEIF ( ( (T1FF -ALPH1M/DELT1 ) .LE. T ) .AND.
&          ( T .LE. T1FF ) ) THEN
           Z1 = DELT1
   ELSE
           Z1 = 0.0
   ENDIF
```

```
 ELSE
    T1FPP = ZETA1/OMGA1M+ALPH1M/DELT1+OMGA1M/ALPH1M
    IF(( 0.0 .LE. T ) .AND. ( T .LT. (ALPH1M/DELT1) ) )THEN
           Z1 = DELT1
    ELSEIF ( ( (ALPH1M/DELT1) .LE. T )  AND.
&         ( T .LT. (OMGA1M/ALPH1M) ) ) THEN
           Z1 = 0.0
    ELSEIF ( ( (OMGA1M/ALPH1M) .LE. T ) .AND.
&         ( T .LT.  (ALPH1M/DELT1+OMGA1M/ALPH1M) ) ) THEN
           Z1 = -DELT1
    ELSEIF ( ( (ALPH1M/DELT1+OMGA1M/ALPH1M) .LE  T ) .AND.
&    ( T .LT. (T1FPP-(ALPH1M/DELT1+OMGA1M/ALPH1M) ) ) ) THEN
           Z1 = 0.D
    ELSEIF ( ( (T1FPP -(ALPH1M/DELT1+OMGA1M/ALPH1M) ).LE.T)
&         .AND. ( T .LT. (T1FPP-OMGA1M/ALPH1M) ) ) THEN
           Z1 = -DELT1
    ELSEIF ( ( (T1FPP-OMGA1M/ALPH1M) .LE. T )
&         .AND. ( T .LT. (T1FPP-ALPH1M/DELT1) ) ) THEN
           Z1 = 0.0
    ELSEIF ( ( (T1FPP-ALPH1M/DELT1) .LE. T )
&         .AND. ( T .LE. T1FPP ) ) THEN
           Z1 = DELT1
    ELSE
           Z1 = 0.D
    ENDIF
    T1F = T1FPP
 ENDIF
 !
 END
```

Jerk Switch Time Generator for Joint 2

```
 SUBROUTINE OPT2(Z2,ZETA2,DELT2,T,TH2IC,DMGA2M,ALPH2M,T2F)
 IMPLICIT NONE
 REAL*4 Z2
 REAL*4 ZETA2
 REAL*4 DELT2
 REAL*4 T
 REAL*4 TH2IC
 REAL*4 DMGA2M
 REAL*4 ALPH2M
 REAL*4 T2F
 REAL*4 T2FP
 REAL*4 T2FPP
 IF ( ZETA2 .LT. ( 2.D*ALPH2M**3/(DELT2*DELT2) ) ) THEN
    T2F = 2.D*( 4.D*ZETA2/DELT2 )**(2.0/3.0)
    IF ( ( 0.0 .LE. T ) .AND  ( T .LT. (T2F/4.D) ) ) THEN
          Z2 = DELT2
    ELSEIF ( ( (T2F/4.0) .LE. T ) .AND.
&         ( T .LT. (3.D*T2F/4.0) ) ) THEN
          Z2 = -DELT2
    ELSEIF ( ( (3.D*T2F/4.D) .LE. T ) .AND.
&          ( T .LE. T2F ) ) THEN
          Z2 = DELT2
    ELSE
          Z2 = D.D
    ENDIF
 ELSEIF( ( ZETA2 .GT. ( 2.D*ALPH2M**3/(DELT2*DELT2) ) ).AND.
&(DMGA2M.GT.(SQRT(ALPH2M**4/(4.D*DELT2*DELT2)+ALPH2M*ZETA2)
&-ALPH2M**2/(2.D*DELT2) ) ) ) THEN

T2FP=ALPH2M/DELT2+SQRT((ALPH2M/DELT2)**2+4.0*ZETA2/ALPH2M )
    IF(( D.D .LE. T ) .AND. ( T .LT. (ALPH2M/DELT2) ) )THEN
          Z2 = DELT2
    ELSEIF ( ( (ALPH2M/DELT2) .LE. T ) .AND.
&          ( T .LT. (T2FP/2.D-ALPH2M/DELT2 ) ) ) THEH
          Z2 = 0.0
    ELSEIF ( ( (T2FP/2.D-ALPH2M/DELT2 ) .LE. T ) .AND.
&          ( T .LT. (T2FP/2.D+ALPH2M/DELT2 ) ) ) THEH
          Z2 = -DELT2
    ELSEIF ( ( (T2FP/2.D+ALPH2M/DELT2 ) .LE. T ) .AND.
&          ( T .LT. (T2FP -ALPH2M/DELT2 ) ) ) THEN
          Z2 = D.D
    ELSEIF ( ( (T2FP -ALPH2M/DELT2 ) .LE. T ) .AHD
&          ( T .LE. T2FP ) ) THEN
          Z2 = DELT2
    ELSE
          Z2 = D.D
    ENDIF
```

```
 ELSE
    T2FPP = ZETA2/OMGA2M+ALPH2M/DELT2+OMGA2M/ALPN2M
    IF ( ( D.D .LE. T ) .AND. ( T .LT. (ALPH2M/DELT2)))THEN
          Z2 = DELT2
    ELSEIF ( ( (ALPH2M/DELT2) .LE. T ) .AND.
&         ( T .LT. (OMGA2M/ALPH2M) ) ) THEN
          Z2 = 0.0
    ELSEIF ( ( (OMGA2M/ALPH2M) .LE. T ) .AND.
&         ( T .LT.  (ALPH2M/DELT2+OMGA2M/ALPN2M) ) ) THEN
          Z2 = -DELT2
    ELSEIF ( ( (ALPH2M/DELT2+OMGA2M/ALPN2M) .LE. T ) .AND.
&    ( T .LT. (T2FPP-(ALPH2M/DELT2+OMGA2M/ALPN2M) ) ) ) THEN
          Z2 = 0.0
    ELSEIF( ( (T2FPP -(ALPH2M/DELT2+OMGA2M/ALPN2M)  ) LE.T)
&         .AND. ( T .LT. (T2FPP-OMGA2M/ALPH2M) ) ) THEN
          Z2 = -DELT2
    ELSEIF ( ( (T2FPP-OMGA2M/ALPN2M) .LE. T )
&          AND. ( T  LT. (T2FPP-ALPH2M/DELT2) ) ) THEN
          Z2 = 0.0
    ELSEIF ( ( (T2FPP-ALPH2M/DELT2) .LE. T )
&         .AND. ( T .LE. T2FPP ) ) THEN
          Z2 = DELT2
    ELSE
          Z2 = D.D
    ENDIF
    T2F = T2FPP
 ENDIF
 |
 END
```

Advanced Continuous Simulation Language Listing for
TOPC with Mid Stream Trajectory Reassignment
for Single-Degree-of-Freedom System

```
PROGRAM 1DOF PLANAR MAN. SYS.,FLDT & OPTIMAL POS CONTROL
"SWITCH1.CSL"
" A PROGRAM THAT IS USED TO TEST NEW SWITCHING ROUTINES"
" REQUIRE LINKING WITN SWITCH1 AND JERK1 SUBROUTINES "
INITIAL
  CINTERVAL CINT = 0.001
  MAXTERVAL MAXT = 0.0001
  NSTEPS    NSTP = 1
  CONSTANT TSTP  = 0.4
  LOGICAL  STUP , BKUP , STGEN
  STUP = .TRUE.
  BKUP = .FALSE.
  STGEN = .FALSE.
  CONSTANT  O1   = 1.0
  CONSTANT  M1   = 10.0
  CONSTANT  G    = 9.8
  CONSTANT  R1   = 100.0
  CONSTANT  RES1 = 1.0
  CONSTANT  INO1   = 0.01
  CONSTANT  A11   = 0.0
  CONSTANT  B11   = 0.0
  CONSTANT  KT1  = 0.5
  CONSTANT  KV1  = 0.5
  CONSTANT  JM1  = 0.002
  CONSTANT TN1IC = -1.0
  CONSTANT OMGA1C = 0.0
  CONSTANT V1 = 0.0
  CONSTANT Z1 = 0.0
  CONSTANT MUL1 = 1.0
  CONSTANT VR1 = 170.0
  CONSTANT CUR1M = 100.0
  CONSTANT OM1M = 340.0
  CONSTANT ALPH1M = 35.0
  CONSTANT TN1F = 0.0
  CONSTANT CLAMPV = 1.5
  CONSTANT CLAMPC = 1.5
  CONSTANT CLAMPA = 1.0
  CONSTANT DELTH = 0.0 , TON = 0.2
  T1F = TSTP
  OMGA1M = VR1 / ( KV1 * R1 )
  DELT1 = ALPH1M* RES1/ ( 4.0* INO1 )
  C11 = KT1*R1
  F11 = -R1*KV1/INO1
  F21 = -RES1/INO1
  LC1 = INO1/(KT1*R1)
  L1 = M1 *G*O1*SIN(TH1)
  CUR1IC = L1 / C11
END $" OF INITIAL "
```

```
DYNAMIC
DERIVATIVE
PROCEDURAL(Z1=T,DELT1,ALPH1M,OMGA1M,ZETA1,STUP,BKUP,STGEN)
CALL CONT1(Z1,T,DELT1,ALPH1M,OMGA1M,ZETA1,STUP,BKUP,STGEN)
END $" OF PROCEDURAL1 "

  ZETA1 = TH1F - TH1IC = DELTH* STEP(TON)

  TH1   = INTEG( OMGA1 ,TH1IC )

   L1 = M1 *G*D1*SIN(TH1)
   J11 = M1*D1*D1 +JM1*R1*R1
   G1P = C11*CUR1 -L1 -A11*TH1 -B11*OMGA1
   G1PP = G1P/ J11
 G1 = BOUND( (-ALPH1M * CLAMPA) ,( ALPH1M * CLAMPA) , G1PP)
   OMGA1 = INTEG( G1 , OMGA1C )

   CUR1D = ( V1/IND1 + F11* OMGA1 + F21* CUR1 )
CUR1 = LIMINT( CUR1D, CUR1IC ,-CUR1M*CLAMPC , CUR1M*CLAMPC )

" BEGINNING OF THE CONTROLLER SECTION "

   Y31 = G1

   L1D = M1*G*D1*OMGA1*COS(TH1)
   J11D = D.D

V1 = BOUND( -VM1*CLAMPV , VM1*CLAMPV , V1P )
V1P=LC1*( J11*Z1+ L1D -F21*L1 +...
(-C11*F11)*OMGA1 +(-F21*J11+J11D)*Y31 )

END $" OF DERIVATIVE "
   TERMT(T GE TSTP)
END $" OF DYNAMIC "
END $" OF PROGRAM "
```

Real Time Decision Controller for
Switch Time Selection and Switch Table Generation

```
 SUBROUTINE CONT1(Z1,T,DELT1,ALPH1M,OMGA1M,ZETA1,STUP,-
&                 BKUP,STGEN)
 IMPLICIT NONE
 REAL*4 Z1,DELT1,ALPH1M,OMGA1M,ZETA1,ZETA0,DEL_ZETA
 REAL*4 CLOCK,CLOCKOLD,T
 REAL*4 T1,T2,T3,T4,T5,T6,T7
 LOGICAL*2 STUP,BKUP,STGEN
 !
 IF ( BKUP .EQ. .FALSE. ) THEN
   IF ( STUP .EQ. .TRUE. ) THEN
   CLOCK = 0.0
CALL SWITCH1(T1,T2,T3,T4,T5,T6,T7,DELT1,ALPH1M,OMGA1M,ZETA1)
   STUP = .FALSE.
   ZETA0 = ZETA1
   ENDIF
   CLOCK = T
   IF ( ( ZETA1 .LT. ZETA0 ).AND.
&        ( CLOCK  GE. (T1+T2+T3+T4+T5) ) ) THEN
      BKUP = .TRUE.
   ENDIF
 CALL JERK1(Z1,T1,T2,T3,T4,T5,T6,T7,DELT1,CLOCK)
 ELSE
    IF ( STGEN .EQ. .FALSE  ) THEN
       DEL_ZETA = -ZETA0 + ZETA1
 CALL
&STATE1(T1,T2,T3,T4,T5,T6,T7,DELT1,ALPH1M,OMGA1M,DEL_ZETA)
       STGEN = .TRUE.
       CLOCKOLD = T
    ENDIF
    CLOCK = T - CLOCKOLD
    CALL JERK2(Z1,T1,T2,T3,T4,T5,T6,T7,DELT1,CLOCK)
 ENDIF
 RETURN
 END
```

```
 SUBROUTINE SWITCH1(T1,T2,T3,T4,T5,T6,T7,DELT1,ALPH1M,
&                   OMGA1M,ZETA1)
 IMPLICIT NONE
 REAL*4 ZETA1
 REAL*4 DELT1
 REAL*4 OMGA1M
 REAL*4 ALPH1M
 REAL*4 T1,T2,T3,T4,T5,T6,T7
 REAL*4 T1F
 !
 IF ( ZETA1 .LT. ( 2.0*ALPH1M**3/(DELT1*DELT1) ) ) THEN
    T1F = 2.0*( 4.0*ZETA1/DELT1 )**(1.0/3.0)
    T1 = T1F / 4.0
    T2 = 0.0
    T3 = T1
    T4 = 0.0
    T5 = T1
    T6 = 0.0
    T7 = T1
 ELSEIF ( ( ZETA1 .GT. ( 2.0*ALPH1M**3/(DELT1*DELT1) )).AND.
& (OMGA1M.GT.(SQRT(ALPH1M**4/(4.0*DELT1*DELT1)+ALPH1M*ZETA1)
& -ALPH1M**2/(2.D*DELT1) ) ) ) THEN
 T1F= ALPH1M/DELT1+ SQRT((ALPH1M/DELT1)**2+4.D*ZETA1/ALPH1M)
    T1 = ALPH1M/DELT1
    T2 = ( T1F - 4.0* T1 ) / 2.0
    T3 = T1
    T4 = 0.0
    T5 = T1
    T6 = T2
    T7 = T1
 ELSE
 T1F = ZETA1/OMGA1M+ALPH1M/DELT1+OMGA1M/ALPH1M
    T1 = ALPH1M / DELT1
    T2 = OMGA1M / ALPH1M - T1
    T3 = T1
    T4 = T1F - 2.0*( 2.0* T1 + T2 )
    T5 = T1
    T6 = T2
    T7 = T1
 ENDIF
 !
 END
```

```
SUBROUTINE JERK1(Z1,T1,T2,T3,T4,T5,T6,T7,DELT1,CLOCK)
IMPLICIT NONE
REAL*4 Z1
REAL*4 CLOCK,DELT1
REAL*4 T1,T2,T3,T4,T5,T6,T7
!
IF ( ( 0.0 .LE. CLOCK ).AND.( CLOCK .LT. T1 ) ) THEN
       Z1 = DELT1
ELSEIF ( ( T1  .LE. CLOCK ).AND.( CLOCK .LT. (T1+T2)))THEN
       Z1 = 0.0
ELSEIF ( ( (T1+T2) .LE. CLOCK ).AND.
&         ( CLOCK .LT. (T1+T2+T3) ) ) THEN
       Z1 = -DELT1
ELSEIF ( ( (T1+T2+T3) .LE. CLOCK ).AND.
&         ( CLOCK .LT. (T1+T2+T3+T4) ) ) THEN
       Z1 = 0.0
ELSEIF ( ( (T1+T2+T3+T4) .LE. CLOCK ).AND.
&     ( CLOCK .LT. (T1+T2+T3+T4+T5) ) ) THEN
      Z1 = -DELT1
ELSEIF ( ( (T1+T2+T3+T4+T5) .LE. CLOCK ).AND.
&     ( CLOCK .LT. (T1+T2+T3+T4+T5+T6) ) ) THEN
      Z1 = 0.0
ELSEIF ( ( (T1+T2+T3+T4+T5+T6) .LE. CLOCK ).AND.
&     ( CLOCK .LT. (T1+T2+T3+T4+T5+T6+T7) ) ) THEN
      Z1 = DELT1
ELSE
       Z1 = 0.0
ENDIF
!
END
```

```
SUBROUTINE JERK2(Z1,T1,T2,T3,T4,T5,T6,T7,DELT1,CLOCK)
IMPLICIT NONE
REAL*4 Z1
REAL*4 CLOCK,DELT1
REAL*4 T1,T2,T3,T4,T5,T6,T7
!
IF ( ( 0.0 .LE. CLOCK ).AND.( CLOCK .LT. T1 ) ) THEN
      Z1 = 0.0
ELSEIF ( ( T1  .LE. CLOCK ).AND.( CLOCK .LT. (T1+T2)) ) THEN
      Z1 = 0.0
ELSEIF ( ( (T1+T2) .LE. CLOCK ).AND.
&        ( CLOCK .LT. (T1+T2+T3) ) ) THEN
      Z1 = DELT1
ELSEIF ( ( (T1+T2+T3) .LE. CLOCK ).AND.
&        ( CLOCK .LT. (T1+T2+T3+T4) ) ) THEN
      Z1 = 0.0
ELSEIF ( ( (T1+T2+T3+T4) .LE. CLOCK ).AND.
&     ( CLOCK .LT. (T1+T2+T3+T4+T5) ) ) THEN
     Z1 = DELT1
ELSEIF ( ( (T1+T2+T3+T4+T5) .LE. CLOCK ).AND.
&     ( CLOCK .LT. (T1+T2+T3+T4+T5+T6) ) ) THEN
     Z1 = 0.0
ELSEIF ( ( (T1+T2+T3+T4+T5+T6) .LE. CLOCK ).AND.
&     ( CLOCK .LT. (T1+T2+T3+T4+T5+T6+T7) ) ) THEN
     Z1 = -DELT1
ELSE
      Z1 = 0.0
ENDIF
!
END
```

```fortran
      SUBROUTINE STATE1(T1P,T2P,T3P,T4P,T5P,T6P,T7P, DELTA ,
     &                  ALPHAMAX , OMAGAMAX , THETA_0 )
      IMPLICIT NONE
C-----------------------------------------------------------------
C  FUNCTION      This subrountine calculates the required
C                  state tables for accessing jerk switching
C                times for time optimal moves.
C
C  AUTHOR        P. MOUNFIELD
C
C  LANGUAGE      FORTRAN-77
C
C  INPUTS          The basic maximum parameters of motion for
C                    the specific system, such as maximum
C                acceleration...
C
C  OUTPUTS         The resulting position and switching times
C                 for incremental delays in the region 6-7 on
C                a general time optimal moves.
C-----------------------------------------------------------------
C                 VARIABLE DECLARATION
C-----------------------------------------------------------------
      INTEGER*4 K                       ! stepping counter
      INTEGER*4 NO_ITERATIONS            ! number of iteration steps
      INTEGER*4 OUT                     ! output unit for data
      LOGICAL*2 FOUND                    ! if true correct ti's were
                                        ! calculated
      REAL*4 T                           ! incremental step in delay
                                        ! time
      REAL*4  DELTA                     ! max jerk
      REAL*4  ALPHAMAX                  ! max acc
      REAL*4  OMAGAMAX                  ! max vel
      REAL*4  THETA_0                   ! original move
      REAL*4  THETA                     ! new move
      REAL*4  T1                        ! time to reach max acc
      REAL*4  DELTA_T                   ! time to reach zero vel orig
                                        ! reg 6-7
      REAL*4  T3                         ! time for move under area
                                        ! delta3
      REAL*4  T32                        ! time moving under con vel
                                        ! area d3
      REAL*4  TCO                          ! time to reach crossover
                                        ! point
      REAL*4  TSTAR                     ! time to reach max vel
      REAL*4  TCV                       ! time moving at con vel
      REAL*4  DELTA_CO                   ! pos moved in tco at con
                                        ! acc
      REAL*4  DELTA_COP                  ! pos moved in tco at con
                                        ! jerk
      REAL*4  W0                        ! initial vel beginning of co
      REAL*4  W1                         ! vel at end of k*T in orig
                                        ! reg 6-7
```

```
 REAL*4  WP                       ! vel at top (bot) of new vel
                                  ! curve
 REAL*4   WPP                         ! const acc vel after
                                  ! crossover
 REAL*4  THETA_0P                 ! absolute pos before reg 6-7
 REAL*4  DELTA_BAR                  ! area under region 6-7 in
                                  ! originial move
 REAL*4  DELTA_BARP                ! area under reg 3-4 with no
                                  ! ov part
 REAL*4  DELTA3                     ! area under region 4-7 in
                                  ! new move
 REAL*4  DELTA2                     ! area under region 2-4 in
                                  ! new move
 REAL*4  T1P,T2P,T3P,T4P,T5P,T6P,T7P
                                  !
CC        10        20        30        40        50
C234567890123456789012345678901234567890123456789
C MAIN
C----------------------------------------------------------
C
C SUBROUTINE STATE1(T1P,T2P,T3P,T4P,T5P,T6P,T7P, DELTA,
C&                 ALPHAMAX , OMAGAMAX , THETA_0 )
 !
 !-----------------------------------------
 ! INITIALIZATION CALCULATIONS
 !-----------------------------------------
 !
 T = .002
 NO_ITERATIONS = 200
 OUT = 10
C      ALPHAMAX = 35.00
C      OMAGAMAX = 3.400
C      DELTA = 875.0
C      THETA_0 = 0.000
 !
 OPEN( UNIT= OUT , FILE= 'STATE1.DAT', STATUS= 'NEW' )
 !
C WRITE(OUT,*)'T=',T,'DELTA=',DELTA,'ALPHAMAX=',ALPHAMAX,
C&                ' OMAGAMAX=',OMAGAMAX
 T1 =  ALPHAMAX / DELTA
 WO = DELTA * T1 * T1 /2.0
 TCO = WO / ALPHAMAX
 TSTAR = ( OMAGAMAX - WO ) / ALPHAMAX
 DELTA_CO =  WO* WO / ( ALPHAMAX* 2.0 )
 DELTA_BAR = DELTA* T1**3 / 6.0
 DELTA_BARP= T1* WO - DELTA_BAR
 THETA_0P = - DELTA_BAR
 !
 K = 0
 FOUND = .FALSE.
```

```
 DO WHILE ( ( .NOT. FOUND ).AND.( K .LE. 500 ) )
    K = K + 1
    !
    !-------------------------------------------------
    ! case a : 0 <= kT < tco
    !-------------------------------------------------
    !
    IF ( ( 0.0 .LE. K*T ).AND.( K*T .LT. TCO ) ) THEN
       !
       !------------------------------------------------
       !
       !------------------------------------------------
       !
       W1 = W0 - ALPHAMAX* K*T
       DELTA_T = SQRT( W1 / DELTA )
       WP = W0 - W1
 DELTA_COP = DELTA* DELTA_T**3 /6.0 + ( W0 + W1 ) *K*T/2.0
 DELTA2 = WP*(T1 - DELTA_T) - DELTA* (T1-DELTA_T)**3/6.0
       T3 = 2.0* SQRT( WP/ DELTA )
       DELTA3 = DELTA* T3**3/ 8.0
       THETA = THETA_OP + DELTA_COP - DELTA2 - DELTA3
C PRINT*,       'case a', THETA, K*T , T1 , T3/2.0 ,T3/2.0
 WRITE(OUT,10)'case a',THETA,K*T,T1,0.0,T3/2.0,0.0,T3/2.0
010  FORMAT( A7,T15,7F10.3 )
C PRINT*,THETA,THETA_0,K
       IF ( THETA .LE. THETA_0 ) THEN
            T1P = T6P + K*T
            T2P = 0.0
            T3P = T1
            T4P = 0.0
            T5P = T3/2.0
            T6P = 0.0
            T7P = T3/2.0
            FOUND = .TRUE.
       ENDIF
       !
```

```
    |----------------------------------------------------
    | case b : tco <= kT < 2*tco
    |----------------------------------------------------
    |
    ELSEIF((TCO .LE. K*T ).AND.( K*T .LT. 2.0*TCO ) )THEN
       |
       |------------------------------------------------
       |
       |------------------------------------------------
       |
       WPP = ALPHAMAX* ( K*T - TCO )
       DELTA2 =DELTA_BARP +WPP*( K*T - TCO )/2.0 + WPP*T1
       WP = WPP + WO
       T3 = 2.0* SQRT( WP/ DELTA )
       DELTA3 = T3* WP / 2.0
       THETA = THETA_OP + DELTA_CO - DELTA2 - DELTA3
C PRINT*,         'case b', THETA, K*T , T1 , T3/2.0 ,T3/2.0
  WRITE(OUT,10) 'case b', THETA, K*T,T1,0.0,T3/2.0,0.0,T3/2.0
       IF ( THETA .LE. THETA_O ) THEN
          T1P = T6P + K*T
          T2P = 0.0
          T3P = T1
          T4P = 0.0
          T5P = T3/2.0
          T6P = 0.0
          T7P = T3/2.0
          FOUND = .TRUE.
       ENDIF
       |
```

```
!
!-----------------------------------------------------------
! case c : 2*tco <= kT < tstar + tco
!-----------------------------------------------------------
!
 ELSEIF((2.0*TCO .LE. K*T ).AND.( K*T.LT. (TSTAR+TCO)) )THEN
     !
     !-----------------------------------------------
     !
     !-----------------------------------------------
     !
     WPP = ALPHAMAX* ( K*T - TCO )
     DELTA2 = DELTA_BARP + WPP*( K*T - TCO )/2.0 + WPP*T1
     WP = WPP + WO
     T32 = ( WPP - WO ) / ALPHAMAX
     DELTA3 = T1* WP + ( WPP + WO )* T32/ 2.0
     THETA = THETA_OP + DELTA_CO - DELTA2 - DELTA3
C PRINT*,       'case c', THETA, K*T, T1, T1, T32, T1
 WRITE(OUT,10) ' case c', THETA, K*T, T1,0.0,T1, T32, T1
     IF ( THETA .LE. THETA_O ) THEN
        T1P = T6P + K*T
        T2P = 0.0
        T3P = T1
        T4P = 0.0
        T5P = T1
        T6P = T32
        T7P = T1
        FOUND = .TRUE.
     ENDIF
     !
```

```
 !
 !----------------------------------------------------
 ! case d : tstar+ tco <= kT < ...
 !----------------------------------------------------
 !
 ELSEIF ( (TSTAR+TCO) .LE. K*T ) THEN
    !
    !--------------------------------------------
    !
    !--------------------------------------------
    !
    DELTA2 =DELTA_BARP + ( OMAGAMAX - WO )*( T1 +
    T32 = ( OMAGAMAX - 2.0* WO ) / ALPHAMAX
    TCV = K*T - TSTAR - TCO
    DELTA3 = OMAGAMAX* ( T1 + TCV + T32/ 2.0 )
    THETA = THETA_OP + DELTA_CO - DELTA2 - DELTA3
C   PRINT*,       'case d', THETA, K*T, T1, TCV, T1,T32, T1
 WRITE(OUT,10) ' case d', THETA, K*T, T1, TCV, T1, T32, T1
    IF ( THETA .LE. THETA_0 ) THEN
       T1P = T6P + TSTAR + TCO
       T2P = 0.0
       T3P = T1
       T4P = TCV
       T5P = T1
       T6P = T32
       T7P = T1
       FOUND = .TRUE.
    ENDIF
    !
  ENDIF
  !
  ENDDO
  !
  RETURN
  END
```

BIBLIOGRAPHY

[1] Ambler, A. P., Popplestone, R. J., and Kempf, K. G., "An Experiment in the Off-line Programming of Robots," 12th International Symposium on Industrial Robots, Paris, France, June 9-11, 1982, pp. 491-502.

[2] Ahmad, S., and Besant, C. B., "Motion Control of Industrial Robots with Closed Loop Trajectories," International Conference on Robotics, Atlanta, Georgia, March 13-15, 1984, pp. 305-311.

[3] Anderson, T. A., and Paul, R., "High Speed Coordinated Control of Industrial Robots," 9th International Symposium on Industrial Robots, Washington D. C., March 13-15, 1979, pp. 449-461.

[4] Brady, M., Hollerbach, J. M., Johnson, T. L., Lozano-Perez, T., and Mason, M. T., Robot Motion: Planning and Control, Cambridge MIT Press, 1982

[5] Castain, R. H., and Paul, R. P., "An On-Line Dynamic Trajectory Generator," International Journal of Robotics Research, Vol. 3, No. 1, Spring, 1984, pp. 68-72

[6] Derby, S., "Off-line Programming of Two Industrial Robots," Robots 8 Conference Proceedings, Detroit, Michigan, June 4-7, 1984, pp. 20-65 - 20-76.

[7] Dillmann, R., "A Graphical Emulation System for Robot Design and Program Testing," Robots 7 Conference Proceedings, Chicago, Illinois, April 17-21, 1983, pp. 7-1 - 7-15.

[8] Dubowsky, S., and Shiller, Z., "Optimal Dynamic Trajectories for Robotic Manipulators," Fifth CISM-IFTOMM Symposium on Theory and Practice of Robots and Manipulators, Udine, Italy, June 26-29, 1984, pp. 96-103.

[9] Flugge-Lotz, I., and Titus, H. A., Jr., "Optimum and Quasi-Optimum Control of Third and Fourth-Order Systems," Technical Report No. 134, Stanford University, Stanford, California, 1962.

[10] Guez, A., "Optimal Control of Robotic Manipulators," Dissertation, University of Florida, Gainesville, Florida, April 1983.

[11] Imam, I., Sweet, L. M., Davis, J. E., Good, M., and Strobel, K., "Simulation and Display of Dynamic Path Errors for Robot Motion Off-Line Programming," Robots 8 Conference Proceedings, Detroit, Michigan, June 4-7, 1984, pp. 4-23 - 4-44.

[12] Jacobs, M. P., "Off-Line Programming: A Current Practical Approach," Robots 7 Conference Proceedings, Chicago, Illinois, April 17-21, 1983, pp. 4-1 - 4-11.

[13] Kahn, M. E., and Roth, B., "The Near-Minimum Time Control of Open-Loop Articulated Kinematic Chains," Journal of Dynamic Systems, Measurement and Control, Vol. 93, September, 1971, pp. 164-172.

[14] Kuno, T., Matsuari, F., Moriba, H., and Ikeda, T., "Robot Performance Simulation," 9th International Symposium on Industrial Robots, Washington D.C., March 13-15, 1979, pp. 323-337.

[15] Leitmann, G., The Calculus of Variations and Optimal Control, New York: Plenum Press, 1981.

[16] Leu, M. C., and Mahajan, R., "Computer Graphics Simulation of Robot Kinematics and Dynamics," Robots 8 Conference Proceedings, Detroit, Michigan, June 4-7, 1984, pp. 4-80 - 4-101.

[17] Lin, C., Chang, P., and Luh, J. Y. S., "Formulation and Optimization of Cubic Polynomial Joint Trajectories for Industrial Robots," IEEE Transactions on Automatic Control, Vol. AC-28, No. 12, December, 1983, pp. 1066-1074.

[18] Meystel, A., "Intelligent Control of a Multiactuator System," IFAC Information Control Problems in Manufacturing Technology, Baltimore, Maryland, 1982, pp. 71-80.

[19] Milenkovic, V., "Computer Synthesis of Continuous Path Motion," Proceeding of the Fifth World Congress on Theory of Machines and Mechanisms, pp. 1332-1335.

[20] Paul, R., "Manipulator Cartesian Path Control," IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-9, No. 11, November, 1979, pp. 702-711.

[21] Patzelt, W., "A Robot Positon Control Algorithm for the Grip Onto an Accelerated Conveyor Belt," 12th International Symposium on Industrial Robots, Paris, France, June 9-11, 1982, pp. 391-399.

[22] Purzley, W., "Robot Path Control by Off-Line Computer," 8th International Symposium on Industrial Robots, Stuttgart, West Germany, April 30 - May 6, 1978, pp. 338-345.

[23] Saridis, G. N., Self-Organizing Control of Stocastic Systems, New York: Marcel Dekker, 1977.

[24] Sjolund, P., and Donath, M., "Robot Task Planning: Programming Using Interactive Computer Graphics," Robots 7 Conference Proceedings, Chicago, Illinois, April 17-21, 1983, pp. 7-122 - 7-135.

[25] Smith, F. W., "Design of Quasi-Optimal Minimum-Time Controllers," IEEE Transactions on Automatic Control, Vol. AC-11, No. 1, January, 1966, pp. 71-77.

[26] Steinmann, J. F., and Klafter, R. D., "Straight Motion of a Manipulator Through Hybrid Computation," Robots 7 Conference Proceedings, Chicago, Illinois, April 17-21, 1983, pp. 2-69 - 20-80.

[27] Taylor, R. H., "Planning and Execution of Straight Line Manipulator Trajectories," IBM Journal Research and Development, Vol. 23, No. 4, July, 1979, pp. 424-436.

[28] Totani, T., and Miyakawa, S., "A Mathematical Model of Hand Transfer Motion for Application to Manipulator Control," Journal of Dynamic Systems, Measurement, and Control, Vol. 102, June, 1981, pp. 152-157.

[29] Toyama, S., and Takano, M., "Study on Speed-Up of Robot Motion," Proceedings of the 11th International Symposium on Industrial Robots, Tokyo, Japan, October 7-9, 1983, pp. 619-628.

[30] Vukobratovic, M., and Stokic, D., "Computer-Aided Control Synthesis of Industrial Manipulators and Its Microprocessor Implementation," 12th International Symposium on Industrial Robots, Paris, France, June 9-11, 1982, pp. 479-489.

[31] Vukobratovic, M., Stokic, D., and Kircanski, M., "Contributions to Dynamic Control of Industrial Manipulators," Proceedings of the 11th International Symposium on Industrial Robots, October 7-9, 1983, pp. 245-252.

BIOGRAPHICAL SKETCH

William Pratt Mounfield, Jr., was born on May 14, 1948 in Roanoke, Virginia, son of Frances Minter Mounfield and the late William Pratt Mounfield. He attended high school at North Cross School in Roanoke, Virginia. Mr. Mounfield has attended the University of Virginia, graduated magna cum laude with a Bachelor of Business Administration from Roanoke College, graduated cum laude with a Bachelor of Science in mechanical engineering and with a Masters of Science in mechanical engineering at Virginia Polytechnic Institute and State University. Mr. Mounfield has also fulfilled his medical school admission requirements at Roanoke College and the University of Virginia.

Mr. Mounfield has worked as a cost accountant for the General Electric Corporation, Salem, Virginia; has pursued advanced skills in manufacturing as a machinist; has worked as a project engineer at Aerospace Research Corporation, Roanoke, Virginia; has worked as an design engineer for the Norfolk & Southern Railway, Roanoke, Virginia; has worked as a design engineer for International Scientific Industries, Christiansburg, Virginia; and is presently on an academic leave as a Drive Systems Engineer from General Electric Corporation, Drive Systems Division, Salem, Virginia. His fields of interest are control systems and robots.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

George N. Sandor, Chairman
Professor of Mechanical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Joseph Duffy, Cochairman
Professor of Mechanical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Robert B. Gaither
Professor of Mechanical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Gary K. Matthew

Gary K. Matthew
Associate Professor of Mechanical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Kermit N. Sigmon

Kermit N. Sigmon
Associate Professor of Mathematics

This dissertation was submitted to the Graduate Faculty of the College of Engineering and to the Graduate School and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

August 1985

Hubert A. Bevis

Dean, College of Engineering

Dean, Graduate School